KTH Industriell teknik och management

Modularisation and product description

- a review and a focused case

David Williamsson

Technical Report TRITA-ITM-RP 2018:5
Department of Machine Design
School of Industrial Engineering and Management KTH Royal Institute of Technology

Summary

This technical report contains the results from the first step of a Ph.D. project in the field of modularisation and product description, conducted by the author at *Scania CV AB* in Södertälje and at *KTH Royal Institute of Technology* in Stockholm.

Scania is one of the leading truck, bus and engine manufacturers in the world and is today a part of the Volkswagen (VW) Group AG, which is one of the world’s largest vehicle manufacturing groups. Scania has a successful history in vehicle modularisation and claims it is one of the most important reasons why they are a leading company today. Scania also has a unique way of describing the modular product in their *generic product structure*, in order to efficiently describe the many product variants. However, the Scania product has over the last years been developed into a mechatronic product with embedded software in focus, demanding the product description as well as the modularisation methodology to support this new dimension. Collaboration within the Volkswagen group also makes it even more important to understand and explain “The Scania Way” of modularising and describing the product.

The purpose of the study presented in this technical reports was to investigate the present state at Scania, concerning product architecting, modularisation, product description and configuration. Hence, this report contains a *literature review*, a *case study* based on semi-structured interviews, as well as an architecture *analysis* of some main Scania components. The analysed main components were chosen to both include mechanical, electrical and software disciplines, in order to highlight some of the challenges when modularising and describing a high-performing product configured from heterogeneous technologies that are developed and managed multidisciplinary. Another purpose of the report was to answer the research questions; what is the present state at Scania, regarding product architecture and management of product data? And what are the unique properties in the modular product architecture at Scania and how are they used, developed and maintained?

The result of the analysis indicates that the nomenclature needs to be further defined at Scania, preferably with a definition which is consistent in order to reduce the risk of confusion and design mistakes during future collaborations. Scania strives to maximise the number of product variants (external variety), while keeping the number of technical solutions low (internal commonality). Hence, a structured methodology which supports the development of the product architecture is clearly needed at Scania, in order to make future collaborations as efficient and successful as possible, and to control the increasing technical complexity in the future Cyber-Physical Systems. Finally, configuration rules are identified to be highly important in order to successfully realise a modular product architecture, since the architecture normally will not be fully uncoupled. A drawback with this approach is that the solution space (i.e. all valid configurations) becomes extremely hard to identify, therefore an advanced product description methodology is essential.

Keywords: *Modularisation, Product Description, Module, Product Structure, Product Architecture.*

NOMENCLATURE

Some of the definitions in the list below are not yet official at Scania (meaning that other definitions may occur), however, the definitions used in this report are commonly used and accepted in the literature (scientific definition). All definitions that do not have the same meaning within literature and at Scania, are highlighted with an "X" in the rightmost column. The reason for this is to highlight areas where the risk for miscommunication potentially is great.

Term	*Scientific definition*	*SCANIA definition*	
Part	Physical unit that cannot be further decomposed, e.g. a screw.	By a part number unambiguously designated identifiable object.	*x*
Component	Simple physical unit, e.g. a pump, which consists of several parts.	Part with a more or less complex, independent characteristic function and intended as part of design.	*x*
Functional element	One of the functions that the product should perform e.g. heat water or reduce drag.	---	
Product Property	Detailed quantifiable statement of what the product has to do.	Product characteristics, e.g. roof height.	*x*
Interface	Surfaces or volumes creating a common boundary between two modules or parts, allowing exchange of signals, energy or material.	Surfaces or volumes creating a common boundary between two parts.	*v*
Modularisation	Identifying the modules for a product, by decomposing it depending on company specific reasons.	---	
Module	Functional building block with standardised decoupled interfaces, which is chosen for company specific reasons.	---	
Component series	---	A series of components divided in different performance steps. The components within the series has the same function.	
Module variant	Alternative of a module with a certain performance or appearance.	---	
Product variant	When combining components in different ways, different variants are created.	When combining components in different ways, different variants are created.	*v*

Performance steps	---	Within a component series, the components may have different performance to satisfy different customer needs. *(see module variant)*	
Product architecture	The arrangement of *functional elements,* the mapping from *functional elements* to *physical components* and the specification of the *interfaces* among interacting physical components.	---	
Product family	Set of products based on the same product platform, which have specific features and functionality to satisfy different customer segments.	---	
Standardisation	Reducing the number of different parts, by identifying and using common parts. Increasing internal commonality.	Same need, identical solution.	***v***
System-Level design	A product's architecture is usually defined during the *System-Level design* (embodiment design), when the functions are examined, arranged and divided into subsystems or modules.	---	
--- = *no official definition exist*			

ABBREVATIONS

At Scania, abbreviations are commonly used in the everyday communication. In the list below, the most common one are mentioned.

A	Part
AI	Part Priming
A-order	Standard order
BOM	Bill Of Materials
CAD	Computer Aided Design
CATIA	CAD-system developed by Dassault Systemes
CEPPSS	Continuous Evolution of Properties Planned in Small Steps
CO	Change Object
DMU	Digital Mock-Up
ECO	Engineering Change Order
Enovia	PLM-system by Dassault Systemes
EP	Enterprise Product
FPC	Functional Product Characteristic
G	Geometry Space
GEO	GEOmetry assurance
GP	Geometry Position
KS	Design structure (Konstruktionsstrukturen)
MECO	Main Engineering Change Order
MPU	Main Product Unit
MU	Module Unit
OAS	Object And Structure (New database platform)
OEM	Original Equipment Manufacturer
PCL	Product Class
PDM	Product Data Management
PLM	Product Lifecycle Management
PU	Product Unit
RCS	Reference Coordinate System
R&D	Research and Development
SOP	Start of Production
SOPS-file	Scania On-board Product Specification
S-order	Special order (outside of the Scania's standard offering)
TCR	Translation Code Register
VCR	Variant Code Register

Scientific abbreviations

DPM	Design Property Matrix
DSM	Design Structure Matrix
IGTA++	Idicula-Gutierrez-Thebeau Algorithm, (DSM clustering algorithm)
MATLAB	Matrix Laboratory, (computing software)
MFD	Modular Function Deployment
MIM	Module Indication Matrix

TABLE OF CONTENTS

1 INTRODUCTION

This technical report was conducted at the department *YMS - PDM & CAD* (Product Data Management & Computer Aided Design) at Scania CV AB and at KTH Royal Institute of Technology in Stockholm, Sweden. The report was the initial step in a modularisation and product description PhD project.

1.1 Background and problem description

Scania is one of the leading truck, bus and engine manufacturers in the world and is today a part of the Volkswagen (VW) Group AG, which is one of the world's largest vehicle manufacturing groups. Scania has a successful history in vehicle modularisation and claims it is one of the most important reasons why they are a leading company today. Scania also has a unique way of describing the modular product in their *product description*, which has a generic product structure in order to efficiently describe the many variants.

The department YMS at Scania, is responsible for the development of PDM & CAD solutions for the entire company, including the R&D department (Research and Development). The YMS department also develops methods and processes concerning the product description, which has a strong connection to the principles of modularisation at Scania.

However, the Scania product has over the last years been developed into a mechatronic product with embedded software in focus, demanding the product description as well as the modularisation methodology to support this new dimension. There is also a growing market regarding offline and online services, which also generates new demands. Collaboration within the Volkswagen group, and employees changing jobs more frequently, makes it even more important to understand and explain "The Scania Way" of modularising and describing the product.

In all business areas within Scania, demands have been raised that the product description must be expanded to include information that is specific for each area. This might result in area-specific new structures; it is clear though that using modularisation in new contexts, has the potential to bring a new dimension to the business' way of working.

The product description could potentially also be used to monitor the modular product architecture and the product complexity, in order to secure that Scania continues to benefit from the many advantages that a modular product enables.

1.2 Purpose and deliverables

The purpose of this technical report was to investigate the present state at Scania, concerning modularisation and product description. The report also contains a literature review and a delimitation process, were some specific main components of the vehicle (in a Scania truck) was chosen to be studied in detail. The main components were chosen to both include mechanical, electrical and software constituents, in order to highlight some of the challenges when modularising and describing a multidisciplinary engineered product in the product description. Different product architecture representations that enable and ease communication between different domain experts were also proposed and piloted in the study.

The following three research questions where addressed:

- *RQ1: What is the present state at Scania, regarding product architecture and management of product data?*
- *RQ2: What are the unique properties in the modular product architecture at Scania and how are they used, developed and maintained?*
- *RQ3: How can we represent the product architecture in general and a modular architecture in particular in a way that facilitates cross-functional communication and collaboration on architecture-related tasks?*

The long term goal with this research project should be aiming towards a mind-set of developing the concept of one common product description that will meet up with the future demands from the market as well as the internal businesses at Scania. One common product description will result in less data saved at multiple sites, which may result in cost savings and increased efficiency as a direct result.

1.3 Delimitations

To fulfil the purpose of the project and to deliver the desired information, without overshooting any deadline, clear delimitations are crucial. Therefore, the following delimitations were identified during the beginning of the project, which were consistent with the project owner demands.

- Only two specific main components will be chosen and studied in detail, i.e. not an entire vehicle.
- The modularisation methodology and product description will mainly be related to R&D and the physical product.
- Only A-orders will be considered when investigating the modularisation process and the product description at Scania, i.e. not S-order with special adaptations.
- Only physical interfaces between technical solutions will be considered.

1.4 Method description

In order to fulfil the purpose of the project, several scientific and industrial methods will be used. Some of the methods will be further explained in the Frame of Reference chapter.

To acquire the latest knowledge within the area of modularisation and product description, an *information retrieval* will first be performed by using the internet, library and meetings with Scania engineers. Thereafter a *Gantt chart* will be created in order to manage the project, which will follow the *Stage-gate process* that will be used as the overall project management tool. The *Gantt chart* will both be visualized in a spreadsheet and by *Visual planning* at Scania, which is a tool to visualize the planning by adding post-it notes in a timeline. The visual planning also includes *pulse meetings* to check the progress of the project and to identify problems.

A functional decomposition of the specific main components in the vehicle will be made by using a *function-means tree*. This tool is used to understand the system (reverse engineering), in terms of function and technical solution, which is essential during modularisation. By later using a *component architecture diagram*, it will be possible to represent the interactions between the components, i.e. the base of the product architecture.

Since this project mainly will involve "desk research", the case study and the research questions will be of an applied nature. Therefore a *qualitative research* approach will be used (Backman, 2016).

The qualitative research approach may look different depending on research project, since it is relatively flexible. However, in this project a general qualitative research process will be used as a guideline, see Figure 1. It should be stated that many of the steps in this process are highly integrated in the practical workflow, and that the illustration therefore only gives a general view of the actual research process.

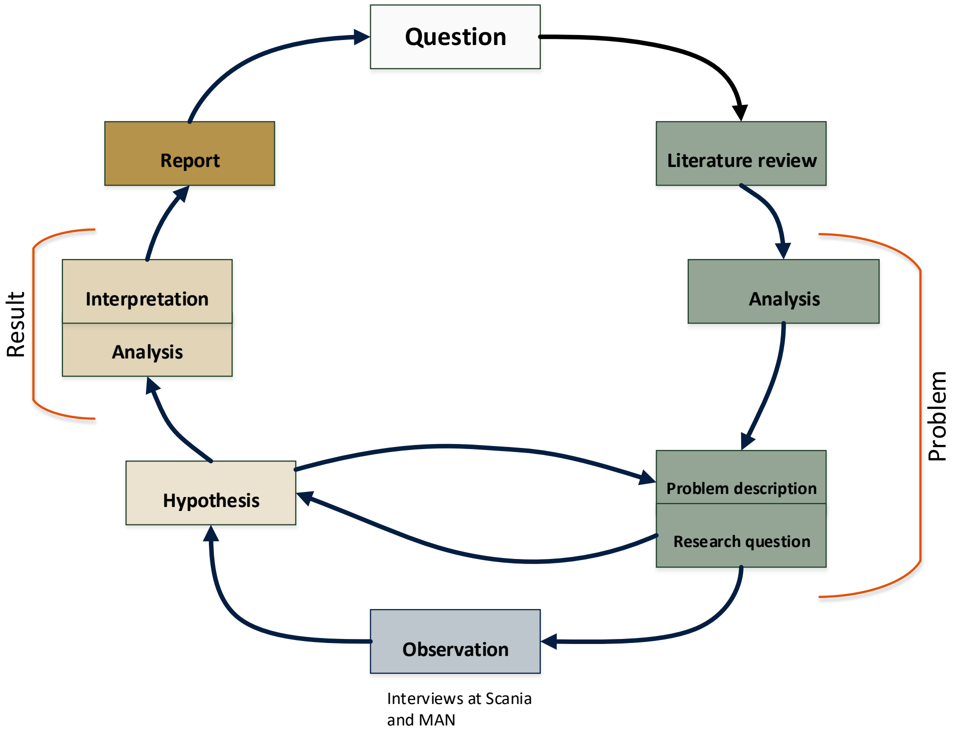


Figure 1. The general qualitative research process (Backman, 2016).

The observational part of the research will be performed with *semi-structured interviews* (Kvale, 1997), to get an insight of the modularisation process, from the perspective of several designers at Scania. The interviews will consist of a set of semi-structured questions, with one respondent at a time. The respondent will be asked to speak openly about one question at a time, followed by supplementary questions based on the answer. The result from the semi-structured interview will later be used to develop hypotheses, explain relations and to create a foundation for the further research within the Ph.D. project.

The methods presented in this section are identified to be the most suitable for the purpose of the project. Several other options are also available, especially modularisation methods. The *MFD* is a common method to modularise products, however it has a strong focus on finding modules based on similarities and according to company strategic reasons. Therefore a more technical approach is also needed, since the product complexity might increase otherwise. The *DSM* method offers that, and in combination with the *MFD* method, it will cover most of the different aspects during the modularisation.

2 FRAME OF REFERENCE

This chapter presents the theoretical reference frame, which forms the basis of the performed research.

2.1 Systems Engineering & Engineering Design

This first chapter explains the context of the following chapters, in the Frame of Reference section. This is done in order to highlight where the presented knowledge is needed and to give a general introduction in the field of Systems Engineering and Engineering Design.

One way to describe *systems engineering* is according to its main function, which is to *manage the engineering of complex systems, in order to add customer value*. Systems engineering therefore selects the path for others to follow, even though there are many other paths available, see Figure 2. Systems engineering also links the different engineering disciplines together, in order to create a complete system.

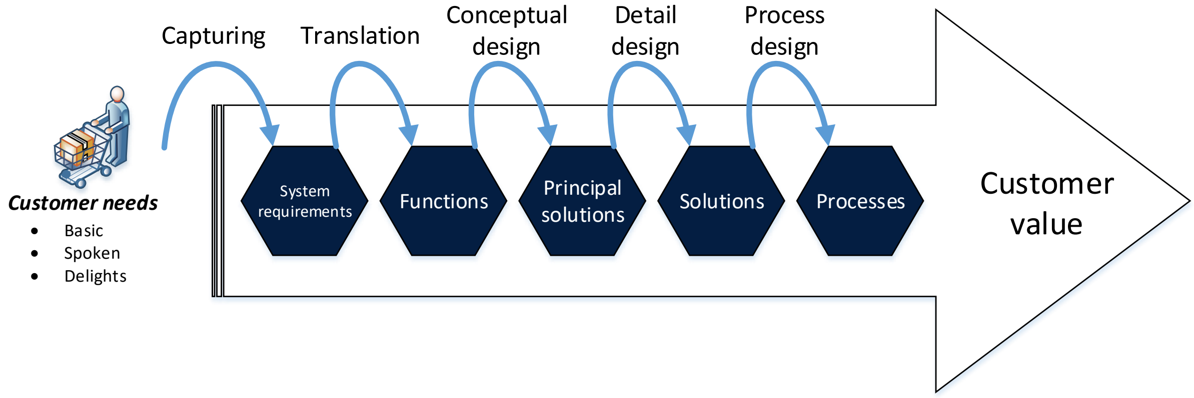


Figure 2. The Systems Engineering Process.

In this report, a *system* is defined as “a set of interrelated elements working together toward some common objective” (Kossiakoff, et al., 2011). As previously defined, systems engineering only deals with complex systems. The term *complex* means that the elements in the system are large in numbers and diverse in terms of type and have many intricate relations with each other. A good example of a complex system is a Scania truck, while a washing machine is a good example of a non-complex system.

One part in the systems engineering life cycle is *engineering design*. The engineering design of a product is a very important part of the product development process, however product development involves much more than design, see Figure 3.

The engineering design process usually consists of three phases; concept development, system-level design and detail design. A very important stage in the system-level design is the creation of the product architecture, which will be further explained in the next chapter. For example, it is the product architecture that determines how the product can be changed during its lifetime, and how the complexity can be handled. Since product architectural decisions are made during the early phases of the development process, where the *R&D* function of a company often plays a lead role, the product architecture is particularly relevant to the R&D function.

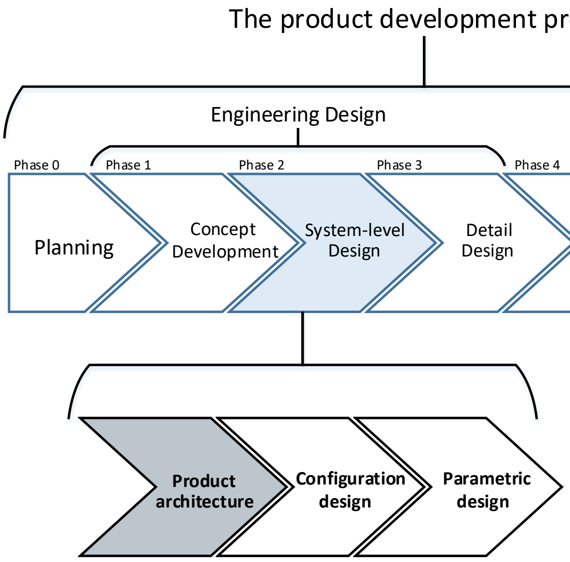


Figure 3. A general product development process.

2.2 Product Architecture

A product's architecture is usually defined during the *System-Level design* (also referred as *embodiment design*), when the functions are examined, arranged and allocated to subsystems or modules.

Ulrich (1993) defined a product architecture as "the scheme by which the function of a product is allocated to physical components." Ulrich also defined it in a more formal way as: the arrangement of *functional elements*, the mapping from *functional elements* to *physical components* (also referred as technical solutions) and the specification of the *interfaces* among interacting physical components.

A functional element is one of the functions that the product should perform e.g. heat water or reduce drag. The arrangement of these functions and the interactions are normally presented in a function structure diagram, which could be a starting point when creating a product architecture (Dieter & C, 2012). The interactions (also referred as relations) are usually described with simple terms e.g. transfer energy.

By mapping the functional elements to physical components, it is possible to see which component or components that are performing each function. If there is a direct dependency between the functional elements to physical components, the design is said to be *uncoupled*, meaning that only one component is performing each function, see Figure 4.

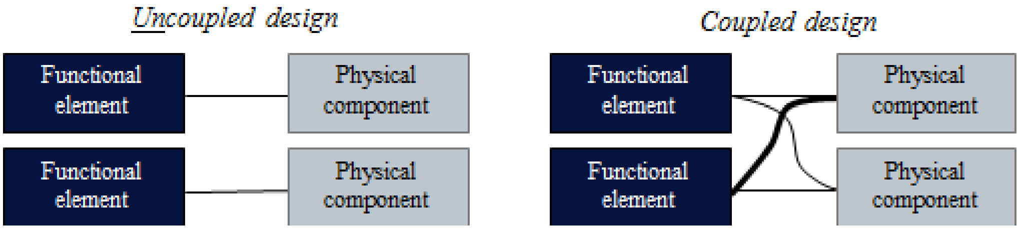


Figure 4. Uncoupled vs coupled design.

In the field of software engineering, the concept of module *cohesion* or *strength* is similar to an uncoupled design.

A practical example of an *uncoupled* design is a modern water tap with a thermostat and a flow control valve, which makes it possible to control the temperature and flow independently, see Figure 5.

In a *coupled* design e.g. an old water tap with two control valves (hot and cold), it is not possible to change the temperature without changing the flow. This makes the controlling of the system unnecessarily complicated. Coupled designs also makes the design or redesign phase harder.

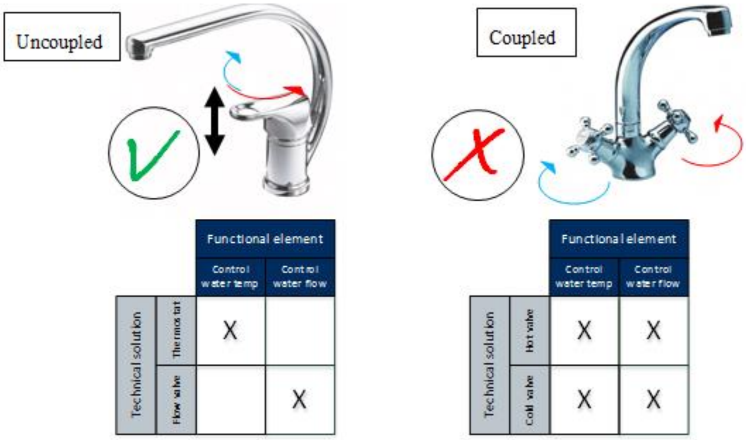


Figure 5. Example of an uncoupled design vs a coupled.

This simple example could easily be transferred to a great variety of examples in a Scania truck, e.g. the control of coolant temperature in the engine. One way to control the flow and temperature out of the engine is with a throttle valve (coupled design) or with a mixing valve in combination with an electrically controlled engine pump (uncoupled).

According to the axiomatic design theory and the *independency axiom*, an uncoupled design is therefore always preferred (Silverstein, et al., 2009), mainly because it is easier to design and control and thus, less complex. An uncoupled design also keeps the design sub-solutions as independent as possible. Therefore, if any part of a system malfunctions, the undesirable consequences do not spread throughout the entire system.

There are two types of product architectures, integral or modular, see Figure 6. In reality, there is also a possibility to have a hybrid architecture, which is a mix of the two types, and commonly there is a degree of modularity. The product architectures will be further elaborated on in the following sections.

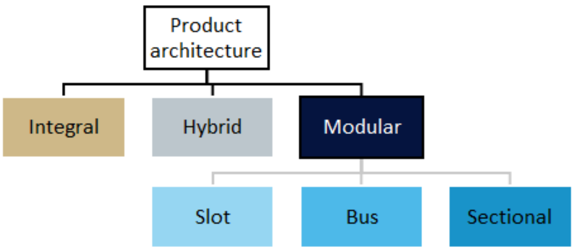


Figure 6. The different product architectures.

Modular product architecture

A modular product architecture should have an uncoupled design, otherwise the modules loses their purpose. It is therefore important to design the product according to the independency axiom.

In order to understand what a modular product architecture is, it is first necessary to define the word *module* (close to Scania internal definition: *component series*). The definition of a module is fairly consistent within the research area of modularisation, however there is sometimes confusion in the industry, since it is loosely used in various types of everyday situations. In this report a *module* is therefore defined as:

1. A functional building block, performing one or several functions.
2. Has specified and standardised decoupled interfaces.
3. Is chosen for company specific reasons.

This definition is consistent with the MFD method and modular product platform definition (Erixon & Ericsson, 1999). Each module could also have different *module variants* (Close to Scania internal definition: *performance steps*), i.e. alternatives of the module with different performance or appearance.

A module should also have *decoupled interfaces*, meaning that it is possible to change between different module variants, or to make a change to one module, without affecting the other modules or how the overall product works (Ulrich, 1993). This is of course very complicated in practice since the interfaces will normally be coupled in some way, for example vibrations from one module will usually be transferred to the rest of the system. However, an interface is still said to be decoupled if the functionality is not affected more than acceptable.

The *interface* of a module is the surface or volume creating a common boundary between two modules, allowing exchange of signals, energy and material.

The interface documentation therefore needs to describe how the modules should interact, which normally is defined with spatial, attachment, command/control and transfer interfaces. It is of course necessary to standardise the interfaces if it should be possible to change between different module variants.

Types of modular product architectures

There are three different types of modular product architectures ,or types of modularity, which are based on their type of interfaces (Ulrich, 1993). Other definitions also occur, however this is one of the most common definitions in the field of Engineering Design.

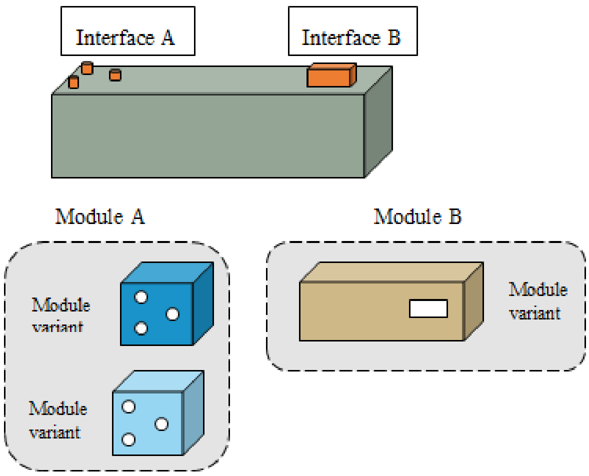


Figure 7. Slot modular product architecture.

The first type of modularity is the *slot* modular product architecture, see Figure 7. In a slot modular product architecture, the interfaces are different between the modules, i.e. interface A

≠ B. However the interface is of course still identical between the module variants within module A or B. This type of modularity can be found in a in a truck dashboard e.g. it is not possible to connect the radio into the speedometer interface.

The second type of modularity is the *sectional* modular product architecture, see Figure 8. In this type of modularity, all modules are connected via identical interfaces, and there is no common base module. An example of this type of modularity is a modular sofa, where different modules can be added to create the desired shape or dimension.

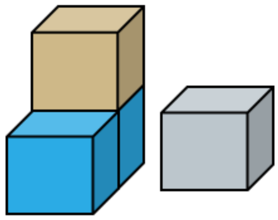


Figure 8. Sectional modular product architecture.

The third type of modularity is the *bus* modular product architecture, see Figure 9. In a bus modular architecture there is a common bus to which the other modules are connected via the same type of interface, i.e. interface A = B. This type of modularity is widely used in computers. As an example, in a USB port it is possible to connect printers and memory sticks etc. to the computer, even though they are performing totally different functions (different modules).

Another example of a bus modular product architecture is the seats in an aircraft, the economy and business class seats (different modules) are attach to the same rails in the floor (common bus). This makes the aircraft highly flexible and allows the airline to change the cabin layout, in order to fit the right customer segments on a specific route.

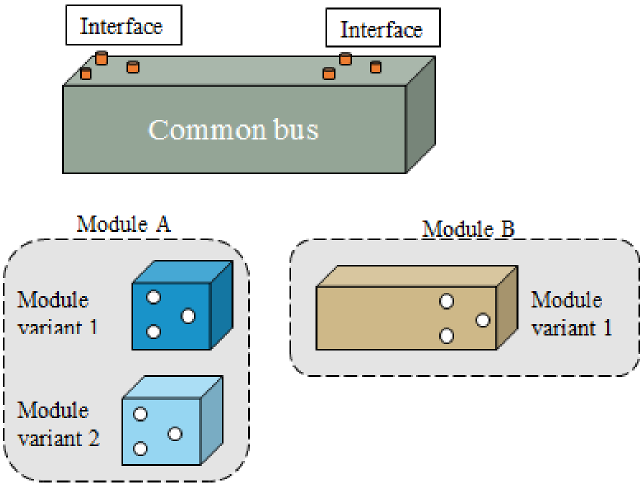


Figure 9. Bus modular product architecture.

Integral product architecture

Sometimes it is not possible or desirable to create an uncoupled or modular design, this normally occurs if the need for high performance is more important than all benefits that a modular architecture can offer.

An integral product architecture has a complex relation between technical solutions and functional elements, as well as coupled interfaces, i.e. the design is coupled (Ulrich, 1993). This makes the design highly complex both when developing, manufacturing and assembling the product, since it is not possible to change one component or function without affecting the others.

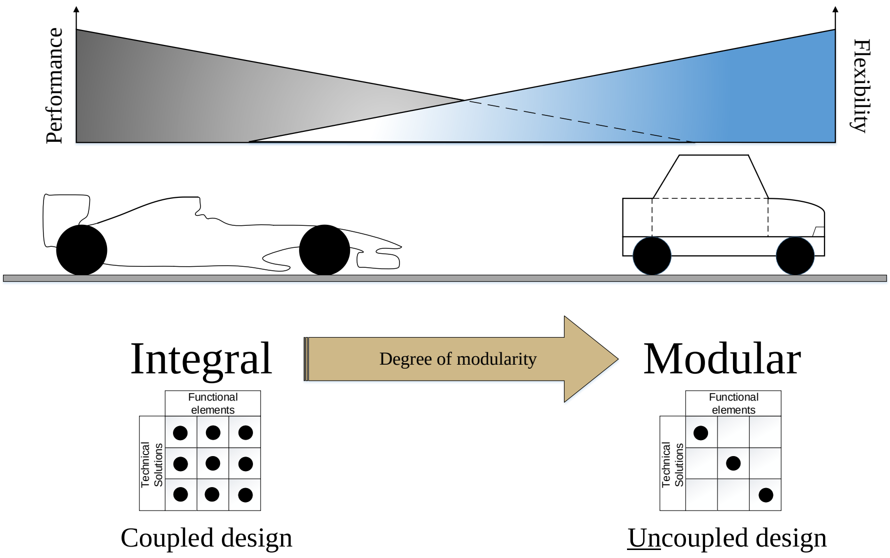


Figure 10. Integral vs. modular product architecture.

An example of a product having an integral product architecture is a Formula one car, see Figure 10. In this type of product, the performance is more important than everything else, which results in an extremely expensive and complex product, but with an impressive performance. A coupled design may therefore be desirable if the need for high performance is most important, e.g. low weight or low volume.

Functional and physical independence

As previously defined, a modular product architecture should have an uncoupled design, i.e. one-to-one mapping between the technical solutions and functional elements. However, if a product should be fully modular, the interfaces must also be decoupled. This means that it should be possible to make a change in one technical solution, without requiring a change in another technical solution, in order for the overall product to perform correctly.

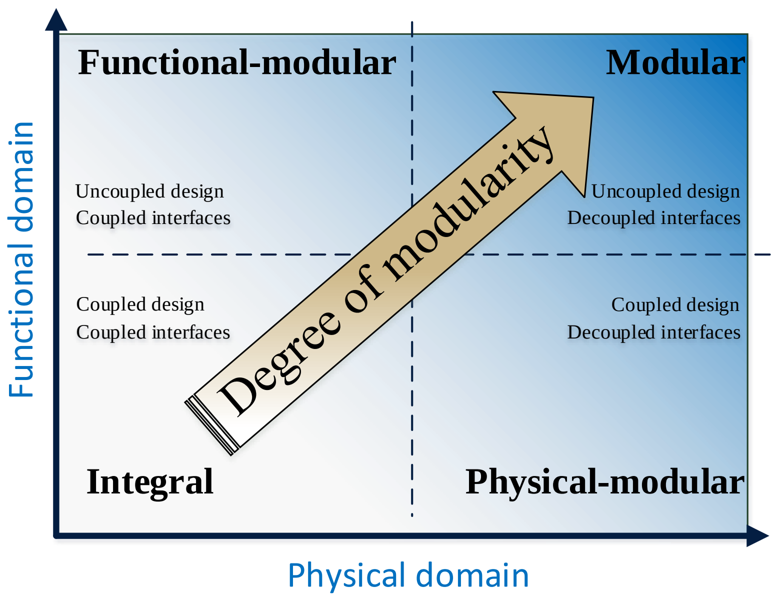


Figure 11. Product architectures based on Göpfert (1998) and (Ulrich, 1993).

In reality, a fully modular or fully integral architecture therefore rarely exists, meaning that almost all architectures are somewhere in between, i.e. there is a degree of modularity. An attempt to expand the classification of product architecture was introduced by (Göpfert, 1998). Besides the integral and modular architecture, two new types were introduced to describe the degree of modularity, see Figure 11.

A *functional-modular* product architecture is defined by functionally independent technical solutions (uncoupled design), which are connected through physical interfaces that are difficult to separate (coupled interfaces). A *physical-modular* product architecture consists of physically independent technical solutions, that are connected through interfaces that are easy to separate, even though the functions are strongly dependent (coupled design). This type of products can easily be disassembled into its modules, but only provide its functionality when the dependent modules are connected.

Examples of coupled and uncoupled interfaces are presented in Figure 12. Please notice that a fully uncoupled design still can lower its degree of modularity by having coupled interfaces. At the same time, a product only having decoupled interfaces can lower its degree of modularity by having a coupled design. Geometrical coupled interfaces are sometimes referred as *geometric nesting*.

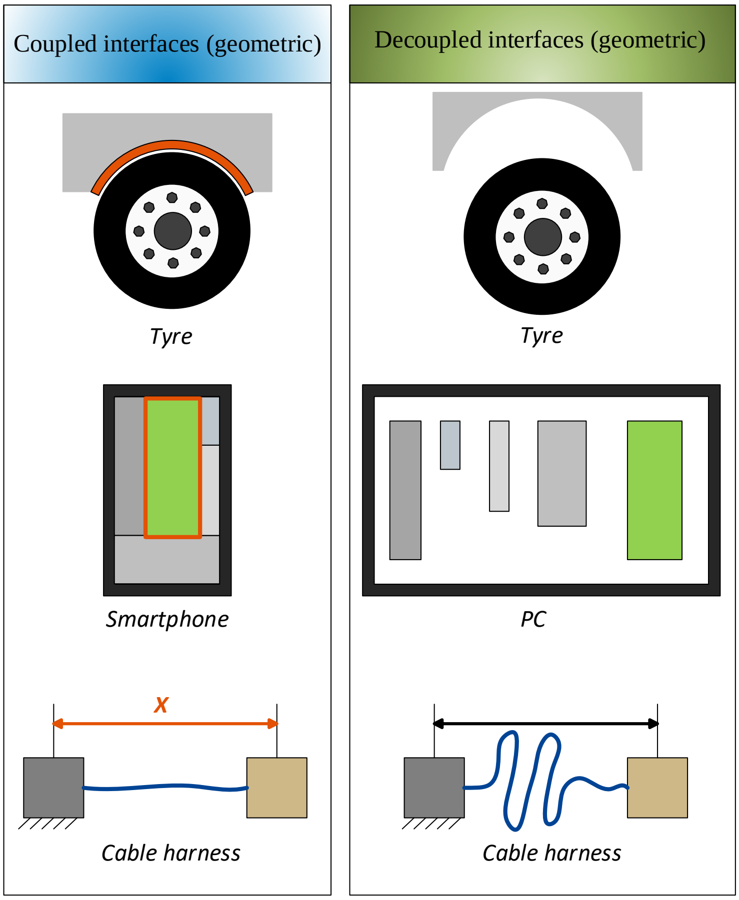


Figure 12. An example of coupled vs. uncoupled interfaces.

An example of a functional-modular product is a smartphone, where the different modules (e.g. battery module and camera module) have one specific function. However, since the modules are tightly packed (geometric nesting), there are many coupled interfaces, see Figure 12. This makes it impossible to change e.g. the battery size without changing the surrounding modules.

In a normal home PC, the different modules (e.g. graphics card) have space around it, which make it easy to upgrade a module without affecting the other modules (i.e. decoupled interfaces). The power supply cable harness to the different modules in the PC, is also longer than it would need to be, for most cases. This allows the interface between the modules and cable harness to be decoupled, i.e. a change in the module performance or location will not require a change to the cable harness. This is obviously not optimized in terms of mass or volume, however the flexibility is high.

A *physical-modular* product architecture could be found in e.g. many types of cars, where the interfaces makes it possible to develop different modules in parallel, and to make a change in one module without affecting the others (decoupled interfaces). The design is however still coupled in many cases, i.e. a module can only provide its functionality when the dependent modules are connected, see Figure 13.

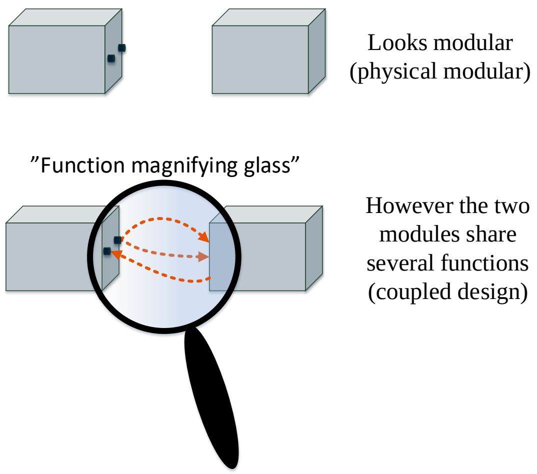


Figure 13. An example of a physical-modular product architecture.

Another example of a physical-modular product is a PC graphics card module with an external “cooling module”, see Figure 14. The internal fan on the graphics card does not have sufficient cooling power in order for the graphics card to provide its functionality. An external cooling module is therefore needed. Hence, the design of this system will be coupled. However, the interfaces are still decoupled, making it easy to redesign the graphics card without changing the cooling module. On the other hand, if the cooling function needs to be changed (e.g. if the customer values change), both the graphics card and the cooling module needs to be changed.

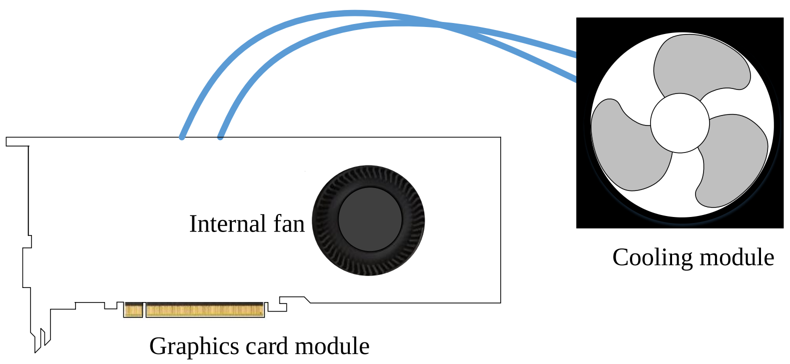


Figure 14. An example of a physical-modular product.

An interface can also be coupled in other ways than geometrical, e.g. through heat transfer and various other physical phenomena. Obviously, all physical interfaces are coupled to some extent. For example, it would not be possible to change the tyre size to Ø5 m, or to increase the surface temperature of a module to 2000 °C, without making substantial changes to the surrounding modules. However, in practical terms, a coupled interface is only relevant to changes that modify the module in some useful way.

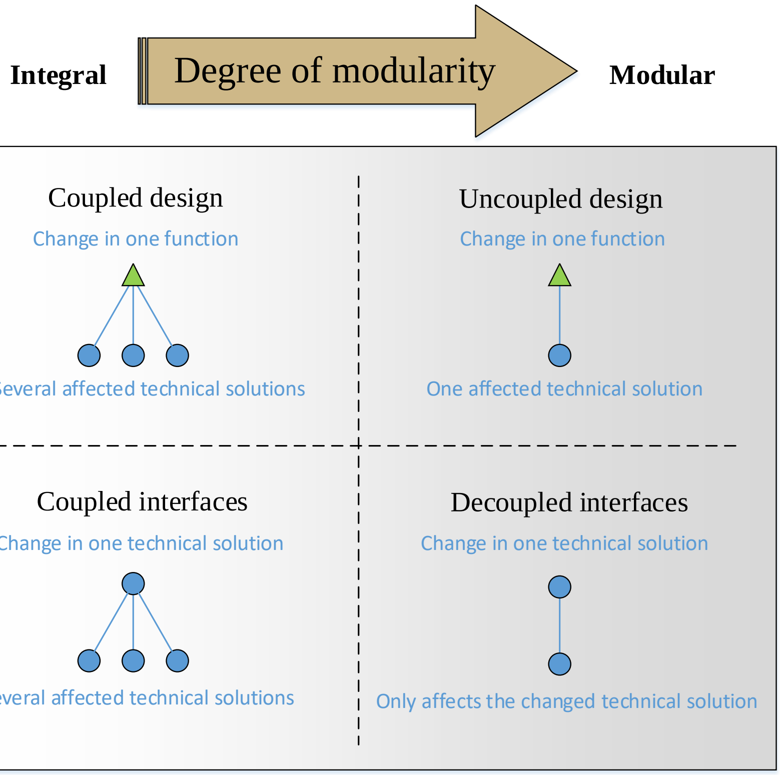


Figure 15. Ease of change to a product architecture.

A product's architecture is therefore closely related to the ease with which a change to a product can be implemented, see Figure 15. Hence, a high flexibility is one of the most important benefits that a modular product architecture can offer.

The global performance of a product is also highly linked to the product architecture, e.g. fuel consumption in a vehicle. A normal design strategy to increase the performance is to apply *function sharing* and *geometric nesting*.

Function sharing is a design strategy in which multiple functions are integrated into a single technical solution, i.e. creating a coupled design. In this way, several technical solutions may be removed, resulting in an increased performance (normally in terms of lower mass). Another design strategy is geometric nesting, where the technical solutions are tightly packed to increase the performance in terms of lower volume.

Product platform & family

A *product platform* is defined as a set of common components, modules or parts from which different products can be efficiently developed and launched (Simpson, et al., 2006). If a set of products are based on the same product platform, and at the same time have specific features and functionality to satisfy different customer segments, the products form a *product family*, see Figure 16.

An example of a product platform is the Volkswagen (VW) A-platform, which consists of floor/chassis modules, drivetrain, and internal cockpit modules. This product platform is shared among a wide variety of Volkswagen brands, e.g. VW, Audi, Seat, and Skoda. All cars containing the same platform form a product family.

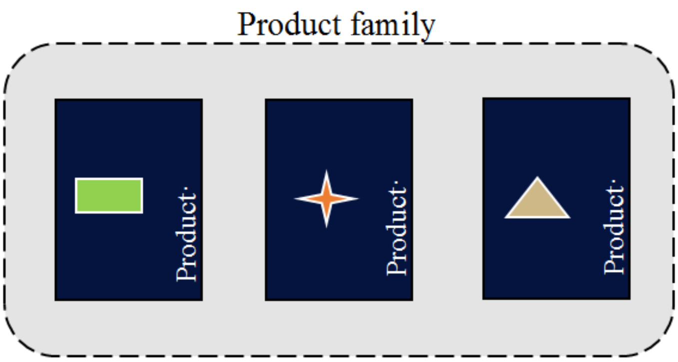


Figure 16. Example of product platform and product family.

There are two kinds of product platforms which could be used to create a product family, see Figure 17.

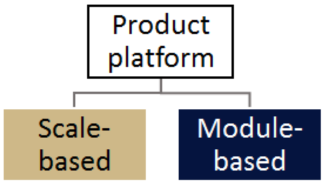


Figure 17. Scale-based and module-based product platforms.

In a *module-based* product platform, one or several modules are added or removed in order to create the desired product platform. It is then possible to add modules or components to the product platform, in order to create the end product. The final product architecture will be modular or a hybrid, depending on if other modules are added or not.

In many types of products, but especially in high performance one with an integral product architecture, *scale-based* product platforms are used.

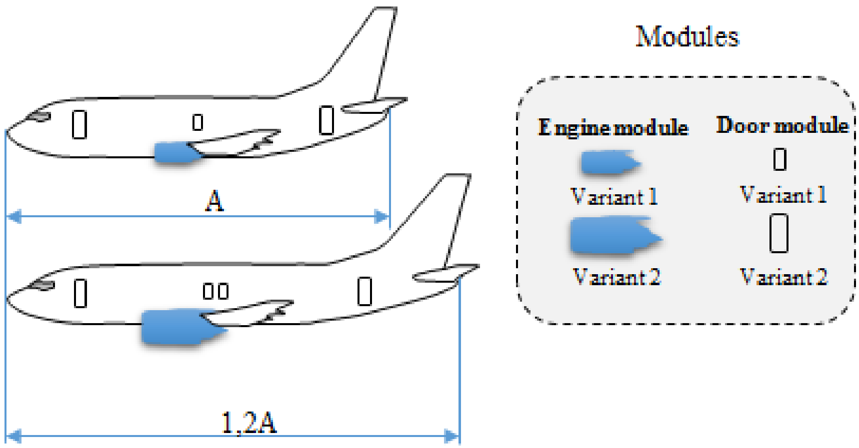


Figure 18. Scale-based product platform, with a hybrid product architecture.

Many aircraft manufacturers therefore use this type of product platform, e.g. Boeing, Airbus and Embraer (Simpson, et al., 2006). By scaling the fuselage and wings (scale-based product platform) of the aircraft, it is possible to create a few different alternatives to the customers, without adding too much development time or manufacturing cost, see Figure 18. An example is the Boeing 777 family, which comes in six variants (Boeing, 2015), with different flight range and seat/cargo capacity.

It is also possible to add modules to a scale based platform, the product architecture will then be a hybrid. This is done in many aircrafts as well, for example the doors and engines might be defined as modules, see Figure 18.

Strategies for modular product architectures

During the 1920s, standardisation and the Taylorism (Scientific Management) enabled an efficient production of the well-known T-Ford. This resulted in a lower manufacturing cost compared with the old tailor-made cars. However the T-Ford customers had no option to decide anything about the car appearance or performance, since everything was standardised, even the colour.

Today the customers and market complexity is far more demanding and complex. At the same time the general product complexity has increased dramatically. The products also needs to be developed much faster in order to beat the competitors. To cope with all these problems, many of the world's most successful companies has modularised their products to some extent. Ford could therefore offer more then 3,8 million different car variants today, in order to fit a large part of the customer segments (Simpson, et al., 2006). At the same time, it does not cost a fortune to get a customised car. The cars are also developed in a much shorter time and with less work effort.

In order to mass produce a product efficiently, an internal commonality of the products is required. However, an external variety is also needed to satisfy different customer segments. Modularisation enables this through "mass-customisation" i.e. mass production and customisation at the same time (Kratochvíl & Carson, 2005).

When modularising a product, the product complexity will also decrease, while the external variety increases. This will result in some of the benefits, but also drawbacks identified in Table 1.

Table 1. Pros and cons by using modularisation, (Erixon & Ericsson, 1999).

Modular product architecture	
+ (Benefits)	**- (Drawbacks)**
High flexibility, easy to redesign some of the modules due to e.g. technology change.	***Reduced product performance***, all module variants will not be optimal in all configurations.
Reduced development time, possible to develop different modules in parallel.	***Brand "cannibalisation"***, products starts to look too similar, which could damage the brand.
Reduced manufacturing time, possible to manufacture different modules in parallel.	***Easier to copy,*** a modular design makes it easier for competitors to copy the design.
Easier administration, efficient product development process.	
Mass-customisation	

The main benefit with a modular product is usually mass-customisation, but also the high flexibility, which allows the product to be redesigned with minimal work effort. This is possible since the modules have standardised and decoupled interfaces, allowing one module to be redesigned without affecting the others.

Another very important feature that modular product architectures offer is the possibility to develop the modules in parallel. This makes it possible to shorten the development time (lead time) dramatically, however it does not necessary mean that the amount of working hours will be reduced. To be able to work in parallel, when developing a complex product, an obvious prerequisite is to define the modules and interfaces. The design teams could then be grouped based on the identified modules, allowing minimal interaction between the teams, in order to develop the product as efficient as possible. As a result, the design teams may consist of people with skills in different technical fields e.g. mechanical and electrical.

Brand cannibalization is one of the drawbacks that might happen if a modular product shares too many modules within the product family. Customers then start to see that they get the same product if they buy the basic product, instead of the high-end.

Even if designing a product architecture is a very complex task, it is highly important since it will affect many aspects of the company and product life cycle. When modularising a product, the modules therefore must be developed to fit the entire company needs and strategies. At the same time, only looking at the strategies when modularising a product might create unnecessarily complex modules and interfaces.

Standardisation vs. modularisation

First of all, it is important to know that modularisation is not standardisation, it is in fact the opposite in terms of product variety and development process, see Figure 19.

Standardisation of a product means that the number of different components are reduced, in order to gain various types of benefits e.g. reduced manufacturing or purchasing cost. However when reducing the number of components, the external variety may decrease to some extent if not handled correctly. Standardisation therefore aims at finding a product that fits all customer segments as good as possible.

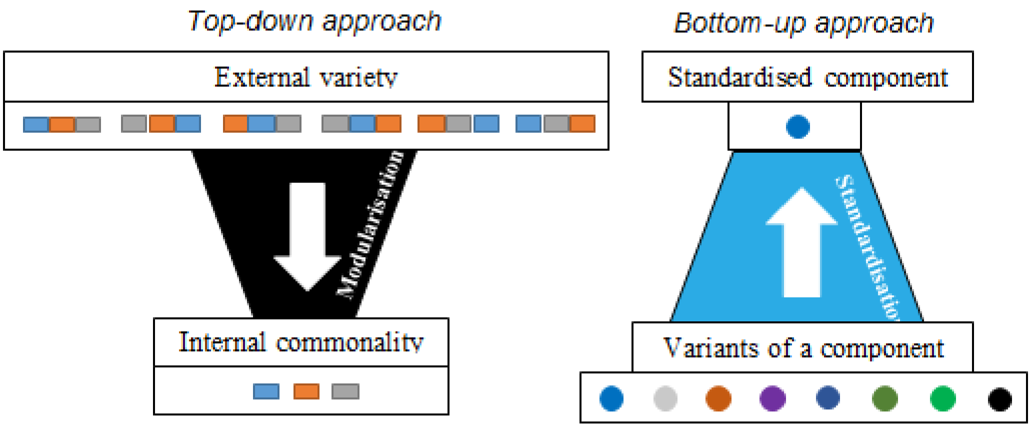


Figure 19. Modularisation vs. Standardisation.

Modularisation of a product means that the product is divided into modules, which could be mass produced to reduce the manufacturing cost and also be pre-assembled to reduce delivery lead time. The modules could, if they are well designed and strategically chosen, be added to create products that fits all customer segments very well (Simpson, et al., 2014). The customers could therefore be offered a product that fit their specific need, at a reasonable price.

Standardisation is a *bottom-up* design approach (Simpson, et al., 2006), i.e. designing from the inside to the outside. This type of process is therefore used when redesigning a product by only looking at the internal component variety, in order to reduce it. As a result, there will be no strategic product plan. An example of standardisation is reducing the number of different screw variants at a company, e.g. only using M12 x 80 mm external Hex-Head screws.

The *Top-down* approach (designing from the outside to the inside) is preferably used during modularisation. This process enables a new or existing product to be developed according to a product plan.

In-house or outsourcing?

Technical products normally contain standard components or subassemblies (commercial off-the-shelf products), that are mass produced and purchased directly from the manufacturer or from distributors. Companies that supply these components or subassemblies are called *suppliers*, and the companies that use these parts in their end products are called *original equipment manufacturers* (OEM). Scania is an example of an OEM, since they develop and sell the end product (e.g. a truck) to the customers, and buys many of the components from suppliers.

A first-tier supplier, is the first supplier in the supply chain to the OEM, which usually means that the supplier is a relatively large company, that is technically capable and takes responsibility for the design and manufacturing.

The decision of making or buying a component (or subsystem) is a very important and usually complex task. For example, the decision will not only impact the cost and quality of the end product. It will also affect the product architecture since a bought component naturally forms a module in the end product. The reason why a bought component logically becomes a module in the end product, is due to its specified interfaces, as well as having a clear function. One of the central parts in the make-buy decision is cost and capability to manufacture the component in-house. If the cost difference between the two alternatives is large, the choice might be easy and obvious. Moreover, if the component involves core technical competency, proprietary manufacturing methods or is critical to the quality of the end product, it needs to be made in-house. However, for many of the components in the end product, it will not be easy to make the choice. To assist the decision, some of the benefits and drawbacks by outsourcing is found in Table 2.

Table 2. Pros and cons by outsourcing (Dieter & C, 2012).

Outsourcing	
+ (Benefits)	**- (Drawbacks)**
Lower manufacturing cost, especially with overseas suppliers.	The supply chain is much longer, delivery delays might cause problems.
Special knowledge in design and manufacturing of the specific component.	Loss of in-house design and manufacturing knowledge. This knowledge may also be transferred to competitors. There is also a risk of reduced quality.
Flexibility in the manufacturing, due to reduced fixed costs.	Hard to improve design for manufacture and assembly (DFMA).
Access to new markets to sell the end product.	If outsourcing to an overseas supplier, several issues may occur, e.g. different language and business culture.

2.3 Modularisation methods

When modularising a product, it is important to first identify a suitable method for the purpose, since there are many modularisation methods available. For very simple products, it might be tempting to only reason about different alternatives in order to find a modular architecture, however this undefined process does not secure that all important aspects are taken care of.

For complex product with many types of systems (e.g. mechanical and electrical), it is simply impossible to find a low complexity, well-functioning, flexible and cost effective modular architecture without the right methods.

The overall aim for all types of modularisation methods is to identify the modules, i.e. the functional building blocks, with standardised decoupled interfaces, which are chosen for company specific reasons. A starting point of any modularisation project will therefore be to have a good understanding of the product and the company strategies. It is important to understand that the modularisation of a product will affect the entire company, including the product development process and the product description, i.e. not only the physical product.

Independently of if a new or existing product should be modularised, it is highly important to have a requirement specification. If a new modular product should be developed, it is important to start thinking about modularity early in the product development process. This enables the design to be uncoupled, which is a good starting point when developing a modular product.

Function-means tree

A function-means tree is typically used during the conceptual design stage, to generate and model the mapping between functions and means i.e. technical solutions, and to explore alternatives. A technical solution could both be a single component or a subsystem, while the function describes what the mean should perform in its most general form e.g. heat water. As earlier described, a modular product architecture should have one-to-one mapping between functions and technical solutions.

The decomposition of a product (Top-down approach) and its representation in a *function-means tree*, is usually a starting point when modularising products. The function and means tree will have a hierarchical structure (Robotham, 2002), see Figure 20.

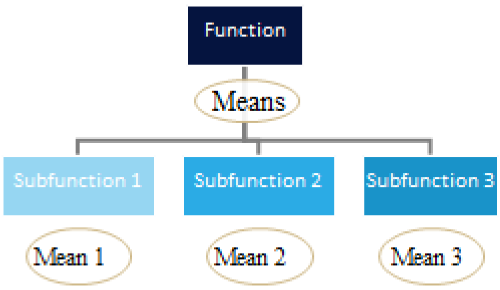


Figure 20. Function-means tree.

The level of decomposition is system dependent, meaning that there is no general way of knowing when to stop the decomposition. However, if the product is decomposed down to every screw and nut, the amount of data will probably be unmanageable and will usually not add any valuable information.

Subsystems that are bought e.g. an electric motor, is also an example when to stop the decomposition. It is of course possible to investigate all the components inside an electric motor, however the important thing when creating the product architecture is not to know the inside of the motor, the important thing is to know that an electric motor is needed to create a function. Bought components/subsystems could therefore be treated as black boxes that realise functions, i.e. they are *function carriers*.

Knowledge integration matrices

A vital starting point when developing any type of product includes deriving suitable *design requirements* from the customer needs. What the customers need and want from a product is usually called *customer requirements*. Examples of customer requirements may be; easy to use, low noise and long service intervals.

The customer requirements must be clearly defined so that a specification of the product can be formulated. One way to accomplish this is to use a simplified version of the Quality function deployment (QFD) method (Erixon & Ericsson, 1999). The QFD method aims at identifying the *voice of the customer,* and to satisfy the customer *needs* and *wants* throughout the entire product development process.

After the customer requirements have been identified, they are translated into product properties in order for the design engineers to know what they should aim at, in order to satisfy the customers. In this report, a product property is defined as a detailed quantifiable designation of the product characteristics. Product properties may sometimes also be referred as engineering specifications in the QFD (Ullman, 2010). Examples of product properties may be; power (W), material rigidity (MPa), weight (kg), time to maintain the system (min) and colour (red, green).

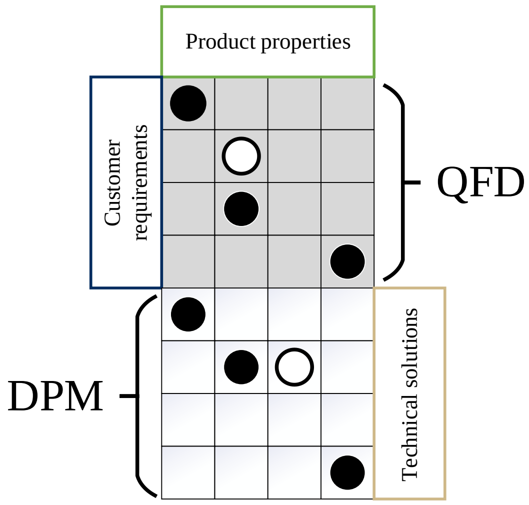


Figure 21. Translation from customer requirements to technical solutions.

The relations (black or partly black bullets) between customer requirements and product properties are inserted into the QFD matrix, see Figure 21. A fully black bullet indicates that there is a strong relation, while an empty filled bullet indicates a weak relation.

Before it is possible to continue with the product design, a *functional decomposition* needs to be performed in order to get a more technical view of the product, and to select the technical solutions. This stage is usually performed with the help of the earlier described functions-means tree. The product properties related to functional performance, are usually created in an iterative process during the functional decomposition.

After performing a functional decomposition, a DPM (Design Property Matrix) can be created to investigate the mapping from *product properties* to *technical solutions*. In this way, it is e.g. possible to investigate how the customer requirements are realised with the technical solutions.

Modular Function Deployment

The MFD (Modular Function Deployment) method was originally developed by Dr. Erixon in his doctoral thesis at KTH Royal Institute of Technology, Stockholm, Sweden (Erixon & Ericsson, 1999). The method was created to take the company strategies and the entire product lifecycle into consideration when modularising a product. This makes the method strongly market driven. However a functional analysis (function-means tree) is also a part of the MFD, to evaluate and improve the design in terms of making it uncoupled.

The MFD method starts with a normal product development method, i.e. defining the customer requirements and transform them into a requirements specification. The core of the MFD method is the MIM (Module Indication Matrix) which is used to identify why different technical solutions (components or subsystems) should become a module.

If using the entire MFD method, it consist of five steps, see Figure 22. The method could both be used to modularise a single product or an entire product family.

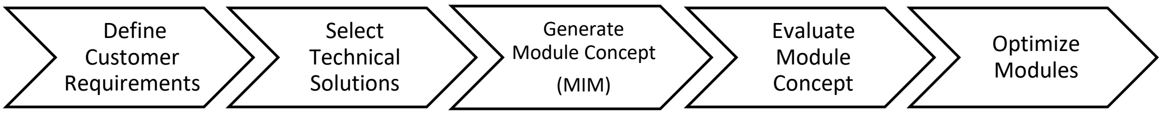


Figure 22. The Modular Function Deployment methodology.

In the MIM, each technical solution is evaluated against the *module drivers*, see Table 3. The evaluation consist of assigning numerical values, 1, 3 or 9, where 9 indicates a strong and 1 a weak driver for the technical solution.

Table 3. The Module drivers.

Module drivers	Description
Carryover	A possibility to reuse the component/subsystem in the next product generation. The component/subsystem is not planned to be developed.
Technology evolution	The component/subsystem will likely be changed due to technology shift or new customer demands.
Planned product changes	The component/subsystem is planned to be changed due to a predefined product plan, e.g. launch of new product models.
Different specifications	It will not be possible to use the same component/subsystem to fulfil all customer demands, within the product family. Variants of the component/subsystem are therefore needed.
Styling	The component/subsystem will be important for the brand and/or will be influenced by trends and fashion.
Common unit	The component/subsystem can be used in the entire product family. A high common unit driver will enable a large production volume.
Process/organization	Components/subsystems that will be manufactured with the same methods/process are suitable to form a module.
Separate testing	Components/subsystems that needs to be tested are suitable to form a module. Enables the entire module to be tested before final assembly.
Supplier availability	The subsystem will be bought directly from a subcontractor, who develops and manufacture the entire module.
Service/maintenance	Component/subsystem must to be easily changed - maintenance/damaged
Upgrading	The component/subsystem will be upgraded in the future.
Recycling	Components/subsystems that are environmentally hostile are suitable to form a module. This enables an easier recycling of the product.

After finishing the MIM, it is important to look for conflicts, i.e. module drivers that are contradicting each other. For example, a technical solution should not be a carryover and be planned for design changes at the same time.

When there are no conflicts in the MIM, it is possible to draw conclusions and find the module candidates by identifying the technical solutions having the highest score in the MIM. Research has shown that there is an optimal number of modules, in terms of assembly time, which occur when the number of modules is equal to the square root of the number of components (Erixon & Ericsson, 1999).

Finally, the rest of the lower weight technical solutions are grouped with the module candidates. There should be no conflicts within the module. If conflicts occur, they need to be resolved by moving the conflicted technical solution to another module.

An example of a MIM can be seen in Figure 23. This example shows how four technical solutions have been assigned numerical values to each module driver. The pump and electric motor got the highest score from the MIM, they therefore become module candidates. The module candidates have then been grouped with other technical solutions having a similar module driver pattern.

		Technical solutions			
Category	**Module drivers**	Pump	Valve	Electric motor	Sensor
Product development and design (R&D)	Carryover	**9**	**3**		
	Technology evolution			**9**	**3**
	Planned product changes				
Variance	Different specifications		3	**9**	**3**
	Styling				
Manufacturing	Common unit				
	Process/organization				
Quality	Separate testing	9			1
Purchase	Supplier availability				
After-sales	Service/maintenance	9		9	
	Upgrading				
	Recycling				
	Total:	**27**	6	**27**	7
		Module 1		Module 2	

Figure 23. An example of the MIM matrix.

One problem with the MFD method is that it does not form modules according to technical complexity, which may result in many drawbacks e.g. complex interfaces.

Design Structure Matrix

Pimmler & Eppinger (1994) originally introduced the DSM (Design Structure Matrix) to represent a product structure, which was later modified to describe a product architecture, by inserting the relations between the technical solutions or functions (Blackenfelt, 2001) into a Product Architecture DSM. The DSM can also be used as a tool for organising a company or processes. The product architecture DSM has been used by many successful companies e.g. when BMW in Germany developed a new concept for a hybrid vehicle architecture (Eppinger & Browning, 2012). The *product architecture DSM* will be referred as DSM in this report, since it is the only type used.

The relations in a DSM can be represented with four types; geometry, signal (information), energy and material, see Table 4.

Table 4. The relations in a DSM.

Type of relation	Two technical solution (or functions) needs to:
Geometry (g)	Be physically connected or orientated to each other.
Signal (s)	Exchange signals (all types) and data between each other.
Energy (e)	Transfer energy between each other.
Material (m)	Transfer material between each other.

By inserting the relations into a matrix, it is possible to represent complex systems in a clear and compact way. It also makes it possible to use clustering algorithms, like the *IGTA++* to calculate module candidates.

The aim is to create clusters (module candidates) that have as many relations as possible within the cluster and as few as possible between them. This allows the interfaces to be as simple as possible, for a given design. As a consequence, products with a coupled design will still have complex interfaces, even if the clustering algorithm finds the best modules for the given design.

As an example of the DSM methodology, the tangle of relations between the components (or functions) in Figure 24 needs a structured representation, therefore they are inserted into the matrix in Figure 25.

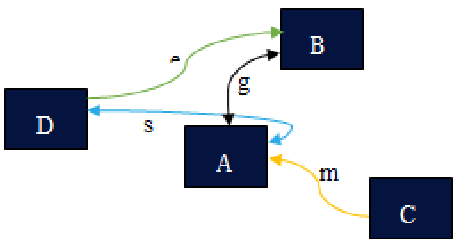


Figure 24. Example of relations between four components.

Please observe that the DSM is not symmetric ($A \neq A^T$), which makes it possible to represent each relation in one or two directions.

		From component			
		A	B	C	D
	A		g	m	s
	B	g			e
To	C				
	D	s			

Figure 25. Example of a DSM, based on Figure 24

Clustering the matrix in Figure 25 (by swapping the rows and columns by hand, or by using a clustering algorithm) yields the matrix in Figure 26.

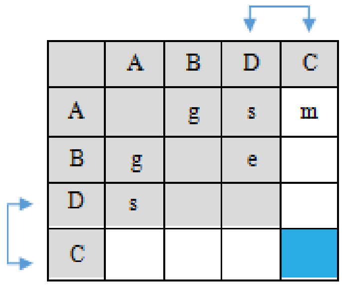


Figure 26. Clustered DSM.

In this simple example, it is easy to see that if the components should be divided into modules, component “C” should be one of the modules since it only has one relation to the others, which makes the interface as simple as possible, see Figure 27. However in large systems with many relations, there is no obvious or easy solution, therefore clustering algorithms plays a crucial role if using a DSM to modularise a product.

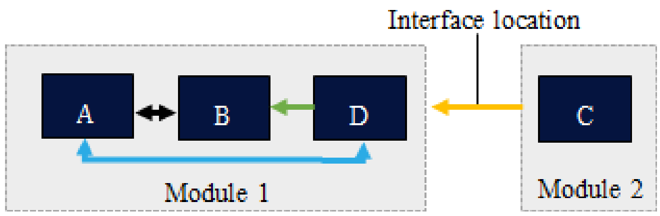


Figure 27. Representation of the modular design.

In this example all relations were equally important, it could however be argued that some relations are more important than others, e.g. a “material” relation could be three times as important as the other relations, since it might be harder to transfer material. Changing the importance of the relations will affect the result of the modularisation and thereby the interface location.

The DSM method enables an objective approach to divide the product into modules. It also enables creativity and new thinking since the module proposals usually are “out of the box”. It does however not take the company strategies into consideration, which the MFD method does.

A new methodology for product modularisation that integrates technical complexity and company strategies was proposed by the author (Williamsson & Sellgren, 2016). The core of the new methodology was to adapt the component-DSM with MIM-strategies, before clustering this hybrid representation with the IGTA++ clustering algorithm. The proposed methodology was exemplified and logically verified with an industrial test rig modularisation case at Scania.

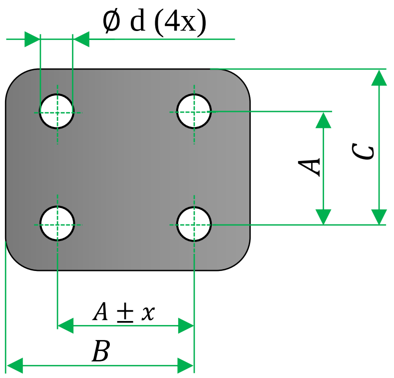
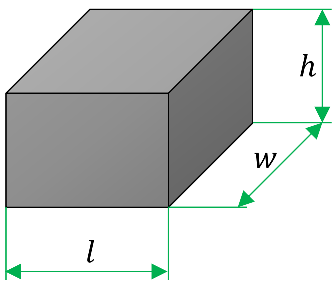
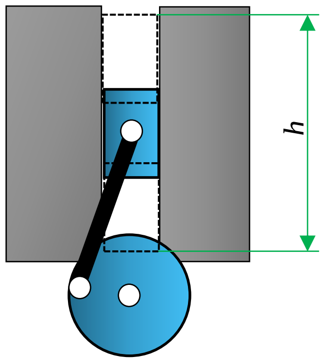
2.4 Module interfaces

An *interface* is the surface or volume creating a common boundary between two modules, allowing exchange of information (usually signals), energy and material (Dieter & C, 2012).

The interfaces should be designed to be as simple and robust as possible. However, if the modules are not carefully designed and chosen with suitable methods, the complexity at the interfaces will be unnecessary high, creating various types of problems and drawbacks. Earlier research has identified that there is a lack of methods to design simple and robust interfaces (Hölttä-Otto, 2005), it is however possible to use normal DFMA (Design for Manufacture and Assembly) knowledge in the design work.

When developing a complex product with many modules, the interfaces will create a communication point between the design teams. It is therefore highly important that the interfaces (and modules) are clearly defined and documented. Since the interfaces are the communication point, wrongly chosen modules will increase the amount of communication between the design teams, resulting in a less efficient product development process.

Table 5. Attachment and spatial interfaces.

<table>
<tr>
<td>
</td>
<td>
</td>
<td>
</td>
</tr>
<tr>
<td>Attachment interface, with specified dimensions and tolerances.</td>
<td>Spatial interface, with specified dimensions.</td>
<td>Spatial interface, the moving piston has a spatial interface against the cylinder.

(There is also a transfer interfaces transferring mechanical energy from the piston to the crankshaft via the connecting rod).</td>
</tr>
</table>

The interface documentation needs to describe how the modules should interact, which is defined according to the following points in this report (Simpson, et al., 2014). Examples of the interfaces can be seen in Table 5, Table 6 and Table 7.

- *Attachment interface*, is a fixed geometric interface between two modules.
- *Spatial interface*, is an interface concerning the space between two modules e.g. volumes or lengths describing the boundaries or movement.
- *Command and control interface*, allows transfer of information (normally signals).
- *Transfer interface*, allows transfer of material or energy.

Some interfaces might follow a standard, e.g. a USB connector. The important thing is however that the interfaces are specified and remains constant over time.

Table 6. Command and control interface.

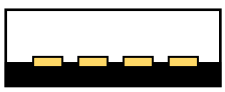
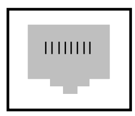
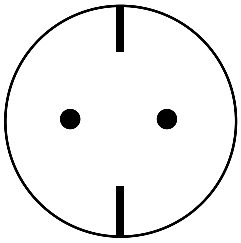
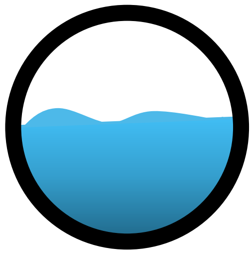
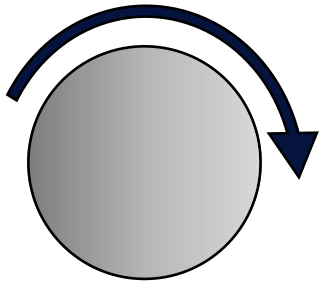
<table>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td>Command and control interface, information (signal) is transferred via a USB connector.</td>
<td>Command and control interface, information (signal) is transferred via an Ethernet connector RJ45.</td>
</tr>
</table>

Table 7. Transfer interface.

		
Transfer interface, energy (electricity) is transferred, e.g. socket CEE 7/4 AC power outlet (230 V).	**Transfer interface**, energy (heat) + material (water) is transferred, e.g. pipe ($\emptyset 40\ mm$) with a cooling water max flow of 200 l/min.	**Transfer interface**, energy (mechanical) is transferred, e.g. rotating shaft ($\emptyset 20\ mm$) with max torque 50 Nm.

The interface documentation is usually a part of the module documentation, which should contain more information i.e. the performance and strategies of the module.

The performance of the interface is also an important feature in the interface documentation. When deciding the interface performance, it is necessary to first know the performance of the entire product and then each module/module variant. It is also important to predict the performance needs for the future. Moreover the performance needs to be evaluated in terms of benefit and cost.

As an example, if a diesel engine and a vehicle cooling system are identified as two modules, developed by two separate teams, the interface documentation will be the only thing that the two teams need to know about each other. The team developing the cooling system are not interested in all performance information about the engine, they only want to know which interfaces that are concerning them, and what the performance should be. From that information, the cooling system team will be able to design their system independently of the other team. Hence, the development will be effective and fast.

2.5 Evaluation methods of modular designs

It has been identified in earlier research (in the area of modularisation) that there is a lack of evaluation tools for modular products (Blackenfelt, 2001). There are however methods that allows some aspects of modular designs to be evaluated. Many of them concerns development time reduction. It is also desirable to investigate other types of benefit that a modular design could offer, these benefits are highly company specific and might be hard to fully predict.

The *development time* or lead time in development, is the time interval from start to finish when developing a product. The development time could be reduced by working in parallel (independently), see Figure 28. There are many benefits by shortening the development time e.g. launching the product before the competitors.

If the design tasks are dependent, the outcome of e.g. task 1 needs to be done before task 2 could be performed, see Figure 28. However if all tasks are independent and uncoupled, there is no need to exchange information between the tasks, which is desirable.

Observe that the *design time* is not constant in Figure 28. The design time is the amount of working hours that is needed to finish a project, and is dependent on the product complexity and how the tasks are coupled.

To be able to work independently, without increasing the design time, the tasks need to be as uncoupled as possible i.e. allowing the teams to work independent with minimal information transfer. Clear definitions and documentation of the modules and interfaces are of course also crucial to make the information transfer as efficient as possible.

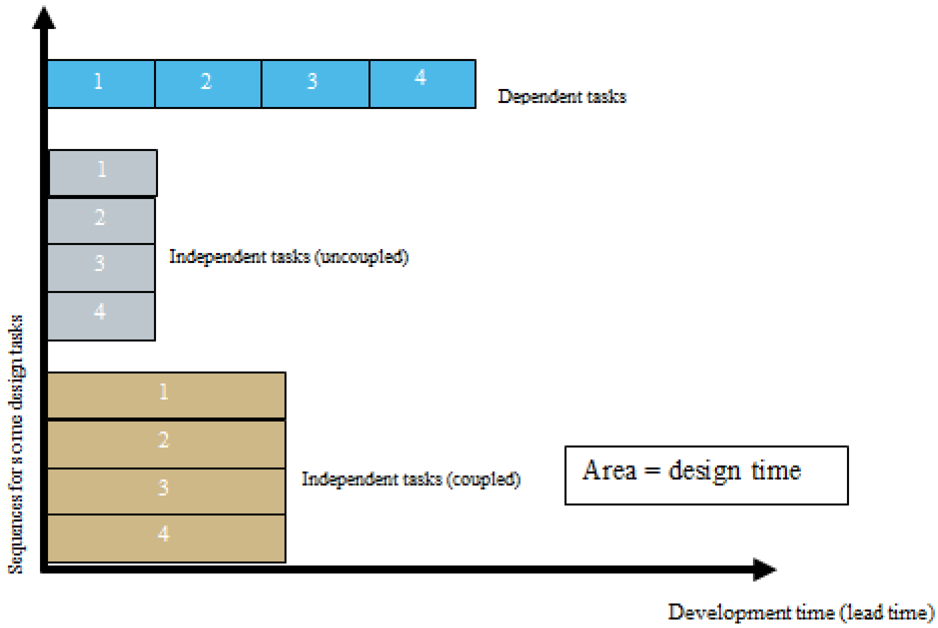


Figure 28. Product development time reduction.

If the design tasks and teams are formed according to the modules identified by the DSM method, the tasks will be as uncoupled as possible. A modular product architecture therefore enables both reduced product complexity, development time and design time. Case studies has shown that it is possible to reduce the development time by 30 - 60%, if changing from an integral to modular product architecture (Erixon & Ericsson, 1999).

The *product complexity factor* is a measure to determine how complex a product is, mainly based on the number of interfaces and modules, see Figure 29.

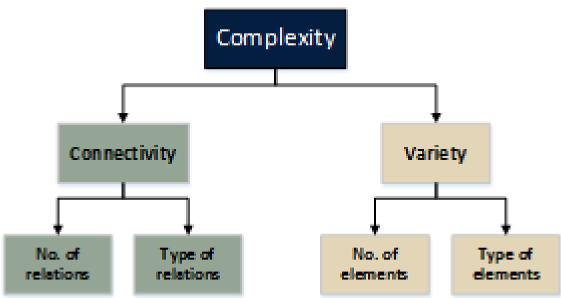


Figure 29. Product complexity.

Since the interfaces are the communication point between the design teams, a low product complexity will result in a short design time, due to few and simple interfaces (Eppinger, et al., 1994). It will also reduce the risk of miscommunication, which otherwise may cause design problems.

In order to determine the product complexity, Pugh (1990) developed a measure for the product complexity factor, see equation (1).

$$\text{Product complexity} = \frac{K}{f} \cdot \sqrt[3]{N_p\ N_t\ \ N_i} \tag{1}$$

Where K is a constant, f is the number of functions, N_p is the number of components or modules, N_t is the number of part types or module variants and N_i the number of interfaces. The number of functions will remain constant if a product is only modularised without making any design changes.

Another important feature that a modular product can offer is the possibility to reduce the *investment* and *development cost* when developing new products. The biggest measure to reduce the development cost is to use carryover modules, i.e. using an old module in a new product. In that way there is no need to spend money on salaries etc. when developing a new module.

The development cost could also be reduced by lowering the product complexity, since the product complexity affects the design time and thereby the salary cost.

Finally, it should be stated that it is possible to use several of the tools found in the modularisation method chapter, in order to evaluate an existing product architecture.

2.6 The Scania Modularisation Legacy

The vehicle industry, in which Scania operates, is a very price sensitive and competitive business in many ways. New technology, competitors, legislation and demanding customers are only some of the things a vehicle manufacturer needs to deal with today. It is therefore very costly to develop and manufacture a vehicle and its components. Many of Scania's competitors therefore focus on volume, i.e. mass production of fairly standardised products. The problem with standardisation, which Henry Ford discovered with the T-ford, is that customers have different needs, everyone is not the same.

Scania realised that it is not possible to create tailor-made vehicles, simply because the volume would be too low, and low volume is not an option since it would not be possible to develop and manufacture the vehicle in an efficient way. They also realised that only selling standardised products (or products with relatively few options) with high volume, would lead to a low profit, also due to the competition. To be highly profitable in this industry, a company clearly needs to mass produce and customise their product at the same time, i.e. "mass-customisation". Scania solved this issue by modularising their product, which led to an external variety (to the customer) and internal commonality (to the entire Scania organisation).

History

The modularisation at Scania started in the 1950s (Scania, 2017), even though some basic principles about reusing components had been identified earlier. At that time, Scania had severe problems with the mechanical design of the components, which resulted in a product that did not fulfil the customer expectations, in terms of quality. To cope with this, one of Scania's newly recruited engineers from KTH Royal Institute of Technology, was given the task to solve the problem.

His name was Sverker Sjöström, and he realised that a component needed to have different performance (e.g. strength) depending on how the customer used the truck (similar to the module driver *different specification* in the MFD method).

He therefore studied the forces acting on the different components during different types of operation, in order to understand how things were dependent. After that, Sjöström realised that e.g. the stress in the axel gear was dependent on the load caused by the gross weight of the vehicle, and the road topography, see Figure 30.

Components having different performance within a *component series*, are today called *performance steps* at Scania (similar to module variants).

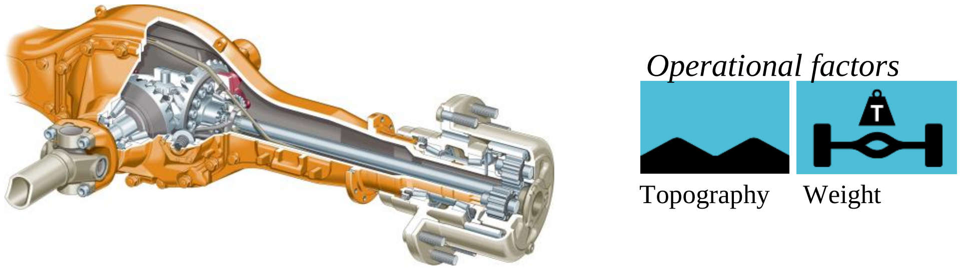


Figure 30. Illustration of a rear axle, and its axle gear, Scania (2017).

In the 1960s, the truck production increased dramatically and Scania started to export a larger part of its products. At the same time, trucks were becoming more complex and customers were more and more demanding. The trucks also had to operate in completely new settings, which resulted in a need for classification of different truck applications, i.e. classification of different types of operation.

Today, the different types of operation are expressed by *operational factors*, which is a way to define the customer's transport task, in order to design and select suitable performance steps from the Scania Modular Toolbox.

Later in the 1960s, the work began with the new cab. With carefully balanced performance steps, in the various component series, Scania could satisfy varying customer needs with a limited number of components. The cabs and frames therefore became a part of the modular system.

In the 1980s, Scania launched its first "fully modularised" truck range, called the GPRT-program, with four cab variants (G, P, R & T), see Figure 31. The program was divided into three operating classes, M (medium duty), H (Heavy duty) and E (Extra Heavy duty), depending on how the truck would be used.

Figure 31. The GPRT-program (the 2-series), Scania (2017).

After introducing the modular system, Scania could meet the changing market demands with simpler development efforts, while achieving economic production at the same time.

The beauty with modularisation was also that each component series, such as transmission, cab and frame, could be further refined by itself, independently of the others (i.e. decoupled interfaces). The principles of modularisation were then continuously used when creating the next product generations, see Figure 32.

Scania claims that modularisation is one of the most important reasons why they are a leading company today, in terms of profitability.

Figure 32. The P-, R- and T-series was introduced in 2004, Scania (2017).

2.7 Modularisation Methodology at Scania

In this chapter, the modularisation methodology at Scania is presented according to how it is intended to be. The information is based on the Scania modularisation book (Scania, 2009), see Figure 33.

At Scania, the modularisation process starts and ends with the customer. This means that Scania modularise their product to fit every customer need as good as possible, in order to make the customer successful and profitable. Scania believes that if the customer makes money, Scania will also make money.

The core of Scania's modularisation principle is carefully *balanced performance steps* with *standardised interfaces*, that can be combined to satisfy different customer needs, with a limited number of components, see Figure 33.

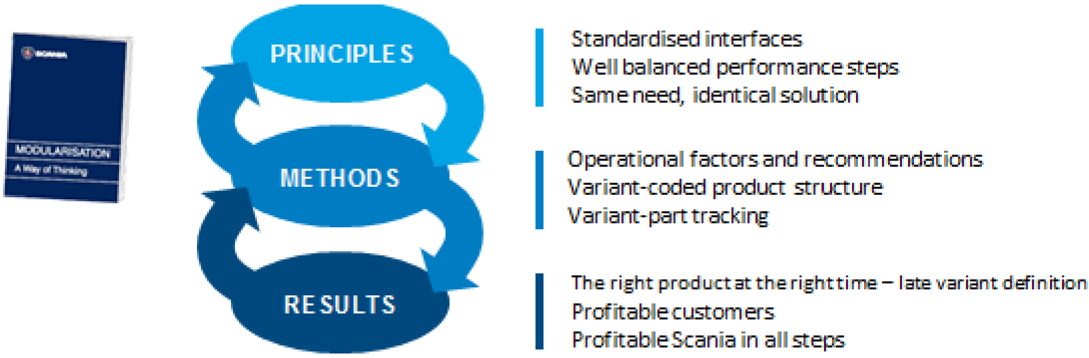


Figure 33. The modularisation process at Scania (Scania, 2009).

The principles

The central part in the modularisation process at Scania are the principles; standardised interfaces, well balanced performance steps and same need, identical solution.

One of the most important principles is the *standardised interfaces*, which creates a common boundary between two component series (modules). The interfaces needs to be clearly specified and have a robust design, making the interface stable over time.

There are three types of interfaces at Scania, see the following bullets and Figure 34.

- *Contact interface*, is a form of attachment and transfer interface.
- *Spatial interface*, same definition as defined earlier.
- *Information interface*, is a command and control interface.

If the interfaces become highly complex, or become hard to standardise, the component series is probably not correctly chosen.

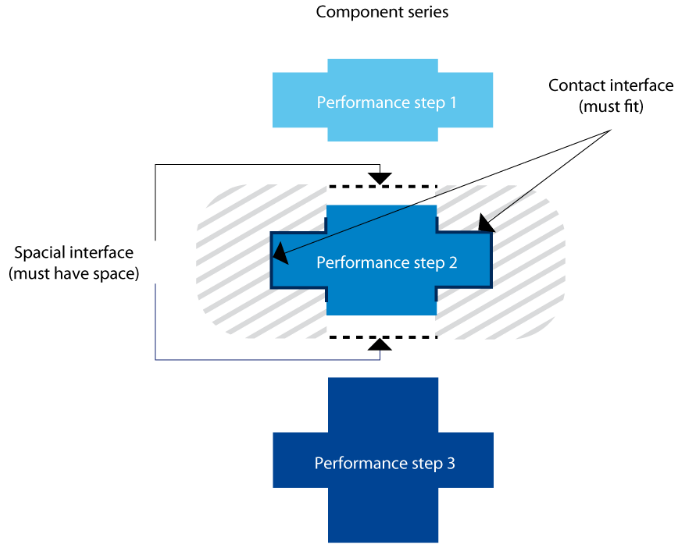


Figure 34. An example of a spatial and contact interface (Scania, 2009).

In order for a function to be realised in a wide performance area, several performance steps within the component series may sometimes be needed. The difficult part is to find a suitable number of performance steps, and which performance each step should have. The goal at Scania is therefore to create *well balanced performance steps*, see Figure 35.

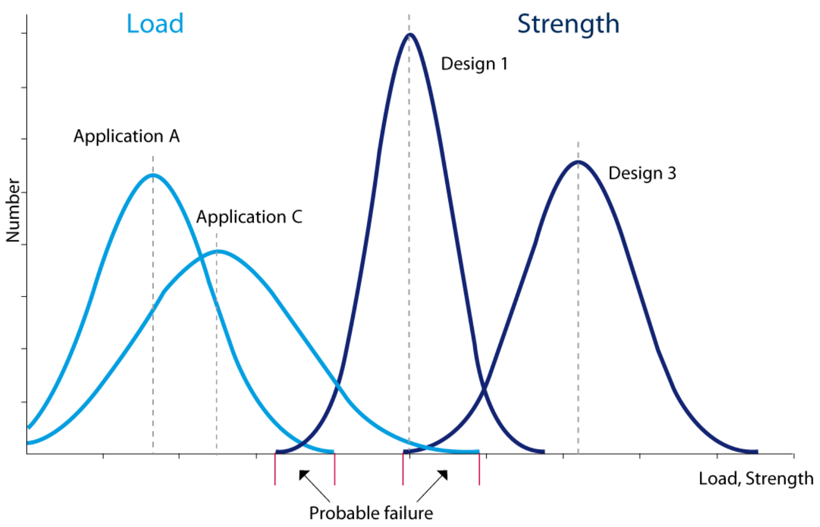


Figure 35. Example of well-balanced performance steps (Scania, 2009).

Except from the technical aspects when choosing well balanced performance steps, the economic aspects also needs to be considered. Every extra performance step adds cost during R&D, production and service, i.e. during its entire life cycle. It is therefore important that there is a clear customer demand for each specific performance step.

The last principle, which also is highly important, is *same need*, *identical solution*. This means that if the customers have the same need, the technical solutions should be identical.

Hence, the same parts should be used in as many performance steps as possible, without affecting the external variety to the customer.

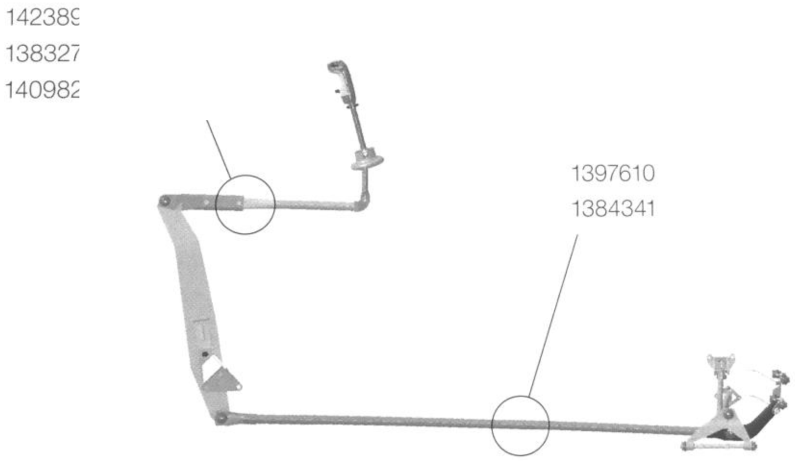


Figure 36. Example of same need, identical solution (Scania, 2009).

An example of same need, identical solution will now be presented, see Figure 36. The mechanical gear shifting component series and its different performance steps originally consisted of 50 different rods (parts). Without changing the external variety to the customer, the number of different parts was reduced to 5, since many of the old parts solved the same need.

One way to simplify the realisation of the principle same need, identical solution, is to have a holistic approach when developing and describing the product, see Figure 37. In the holistic approach, the product is developed and described in a general way (generic), which makes it possible to fit a large part of the customers different needs. In the traditional approach, which is used at many other companies (including Scania in the past), different product platforms are used to fit specific customer segments.

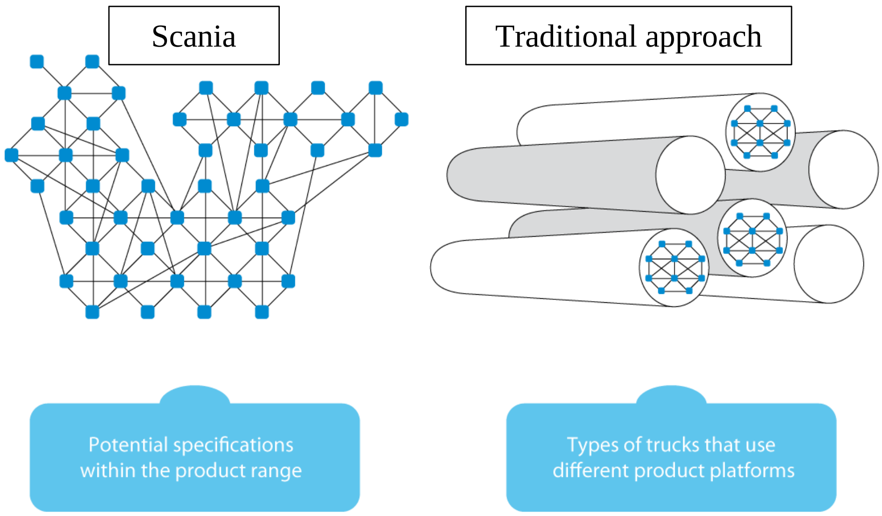


Figure 37. Same need, identical solution (Scania, 2009).

One problem with the traditional customer segment approach is how the company usually is organised and managed. The normal approach is that a project is formed to create one new platform, usually to meet a new customer demand. The members of that project will focus on their project, and will usually not talk with other projects, i.e. it is hard to see the similarities between the projects and its platforms. Companies using this approach may therefore have a large number of parts, but not a very high external variety to the customers.

The methodology

The modularisation methodology at Scania consists of; operational factors and recommendations, variant-coded product structure and variant-part tracking.

Operational factors is a common way to define the customer's transport task, in order to increase the understanding between Scania and the customer, see Figure 39. The operational factors could therefore be seen as the interface between Scania and the customers, in both directions, i.e. both when developing and selling the product.

The operational factors are therefore used to modularise the product and to design the performance steps, as well as guiding the customer to the optimal vehicle configuration.

The *technical recommendations* are mainly used to indicate how to choose between the performance steps under a given set of operational factors, see Figure 38. The recommendations are distributed to the sales and service functions at Scania via e.g. the Vehicle Optimizer, which is a software simulation tool used for examine and compare different vehicle configurations.

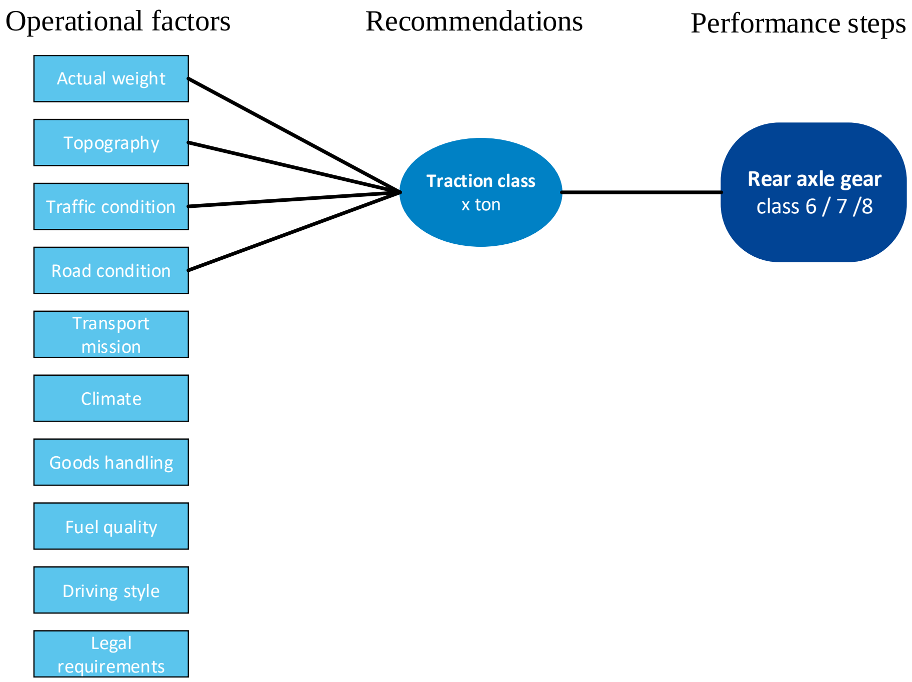


Figure 38. Relation between operational factors and performance steps.

An example of a technical recommendation will now be presented, based on the rear axle gear. *The rear axle gear R560 is suitable for short transport assignments and is recommended if the actual weight of the vehicle is under 18 tonne. This axle gear cannot be combined with a differential lock.*

Figure 39. Operational factors (Scania, 2009).

The *variant-coded product structure* was created to describe the modularised Scania product in an efficient way. The product structure at Scania is therefore *generic*, meaning that it does not describe a single product, but rather the entire product portfolio, i.e. the Scania Modular Toolbox, see Figure 40. Please notice that the Scania Modular Toolbox consists of many more components than illustrated in Figure 40.

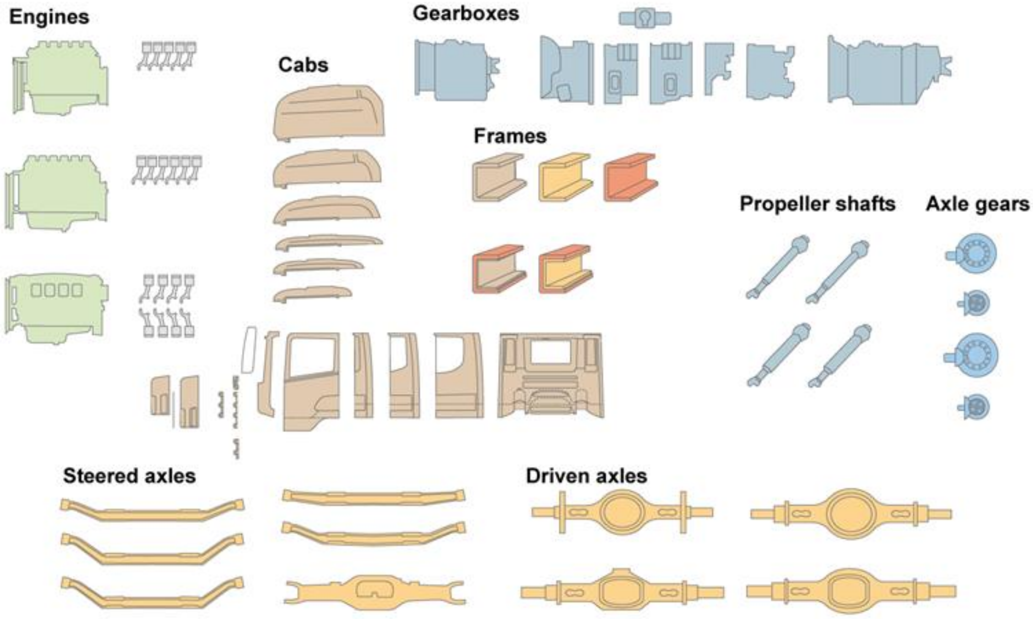


Figure 40. The Scania Modular Toolbox (Scania, 2009).

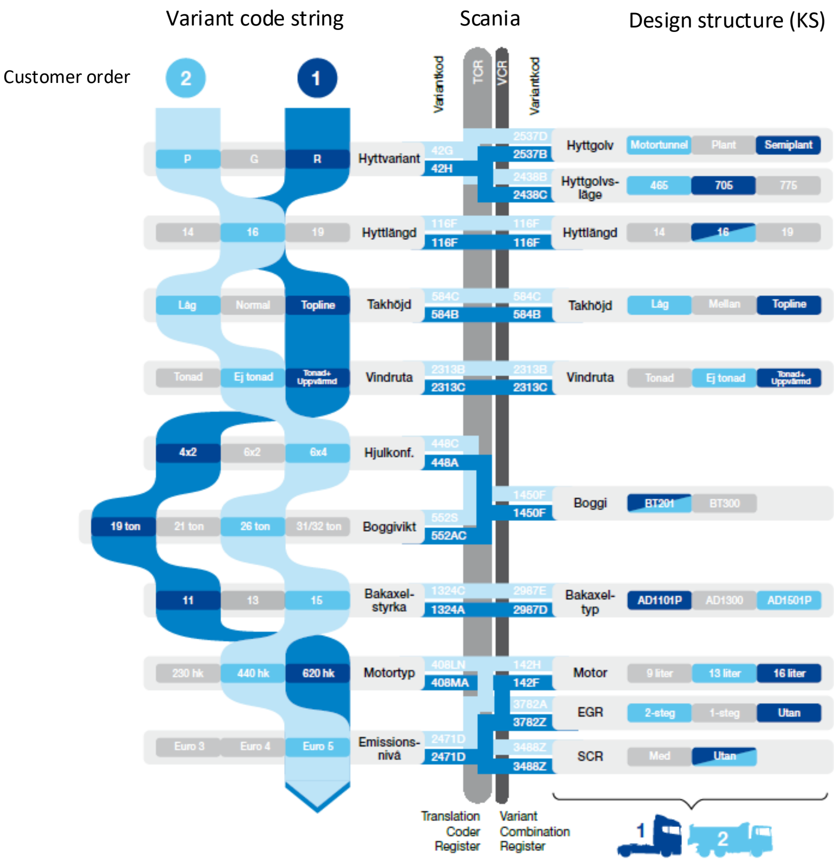


Figure 41. Variant-coded product structure.

In order to describe the many different variants (product configurations) that can be created from the Scania Modular Toolbox, *variant codes* are used, see Figure 41. The variant codes therefore describe which component series and performance steps that should be used in a specific product variant to a customer. The variant codes also describe other properties, e.g. if the steering wheel should be on the left or right-hand side. Hence, the variant codes can be seen as configuration codes that acts as a “key” (or as product “DNA”), which allows a specific product variant to be described from the generic product structure. A component should not be dependent on the choice of another component, as long as it is possible (i.e. decoupled interfaces). Thanks to the variant-coded product structure, a product can easily be configured according to the actual customer demand. In a normal platform based product structure, a specific platform and its components are configured early in the development towards a future or identified customer demand that might be totally wrong when the product is ready to be launched. The Scania product description will be further explained in a later chapter.

To describe the possible customer choices in the Modular Toolbox, a *technical specification* is used. The technical specification also describes restrictions due to selected combinations, e.g. all engine variants cannot be combined with all gearbox variants.

The *variant-part tracking* is used to assist and visualise the modularisation at Scania, see Figure 42. The number of parts is calculated based on the sum of unique part numbers, within the component series or subsystem. The number of variants is calculated based on the possible number of variants (product configurations) within a component series or subsystem. The number of variants could be seen as potential income, while the number of parts could be seen as cost.

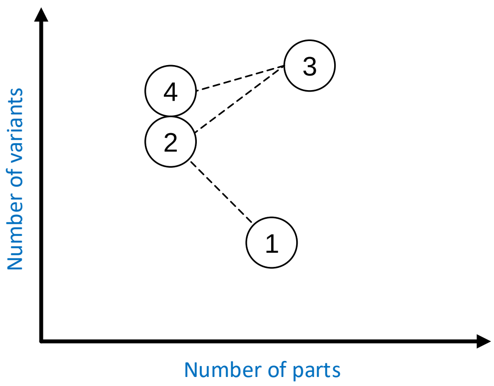


Figure 42. Variant-part tracking.

Methodological analysis

To handle the product variation and customisation efficiently, Scania strives towards *late variant definition*, meaning that the configuration of the component series and performance steps should take place as late as possible, i.e. preferably after the customer has bought the product. It should be stated that late variant definition is an outcome by having an modular product, and that an integral product architecture cannot offer this benefit.

By making the variant configuration as late as possible in the development process and production chain, Scania is able to benefit from many advantages, e.g. improved inventory savings, customer service, and lower overall costs.

This strategy could be found at multiple cases outside of Scania, e.g. the different variants of USB chargers that is needed to fit different power sockets in different countries, see Figure 43.

The *different specification* module should be assembled as late as possible, preferably at the customer, while the *common unit* module should be included to each and every customer. In this way, the common unit may be mass produced. At the same time, the charger is not limited to a certain market, i.e. mass customization.

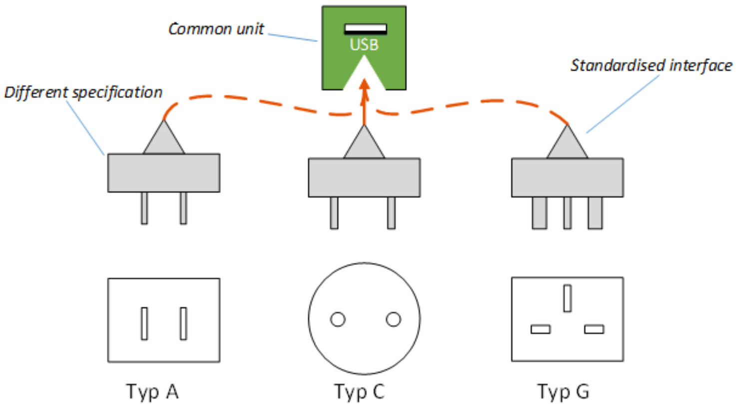


Figure 43. General example of late variant definition.

A similar example is the process when buying a new bicycle. The bicycle does normally not include any extra features (e.g. no kickstand or fenders) when it arrives from the factory. This means that the configuration takes place after the customer has bought the bicycle, resulting in an cost efficient customization process.

One more result by having a modular product at Scania is *profitable customers*. In order for a customer to be profitable, the product needs to fit the specific transport task, resulting in a customer that can perform the task as good as possible, and by that make a good profit. Scania believes that if the customers are profitable, it will result in a *profitable Scania in all steps*. The modularisation also results in an efficient Scania in many ways, for example, new variants can easily be developed with little work effort, i.e. external variety and internal commonality. The modular product also results in quality improvements, mainly since it is possible to make small design changes, one at a time, instead of making one big change.

2.8 Product Description

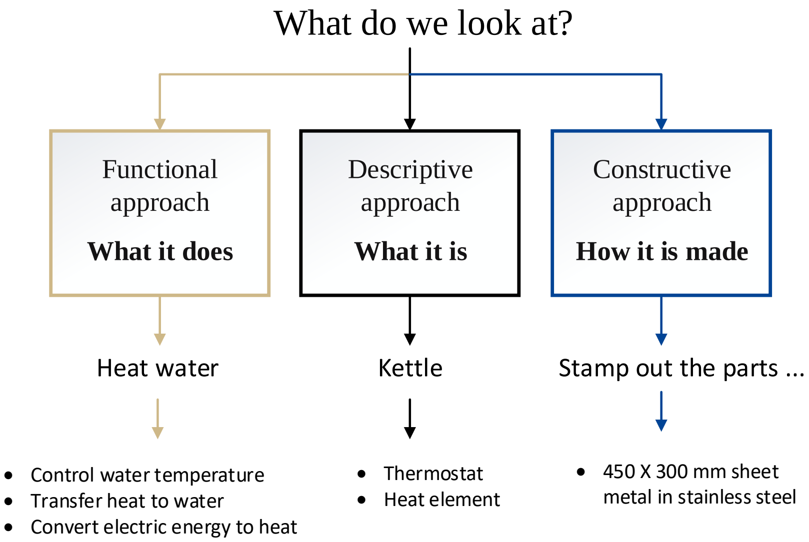


Figure 44. An example of different viewpoints when describing a product.

Depending on which type of product architecture and strategy a company has chosen, the products should be described accordingly. In order for a modular product architecture to be successfully realised in practice, a suitable *PLM system* is therefore absolutely essential.

When describing a product, different viewpoints will exist, see Figure 44. There are three general ways to look at a product; what it *does*, what it *is* and how it is *made*. All these aspects are important in order to describe a product, but are useful for different departments at a company (Foussier, 2006). It should be stated that many other viewpoints of a product also exist and may be important in some cases.

Introduction to PLM and PDM

Product Lifecycle Management (PLM) is a strategy and a systematic method for both managing and developing industrially manufactured products and related information (Saaksvuori & Immonen, 2008). A PLM system therefore manages all product variants, i.e. the entire product portfolio and all of its parts, through the products entire lifecycle, from innovation to recycling. Ideally, a PLM system therefore works as an information processing system that integrates the functions of the entire company, see Figure 45.

The main drivers of PLM can roughly be summarised to the following bullets.

- Increase product revenues.
- Reduce product-related costs.
- Maximise value of the product portfolio.

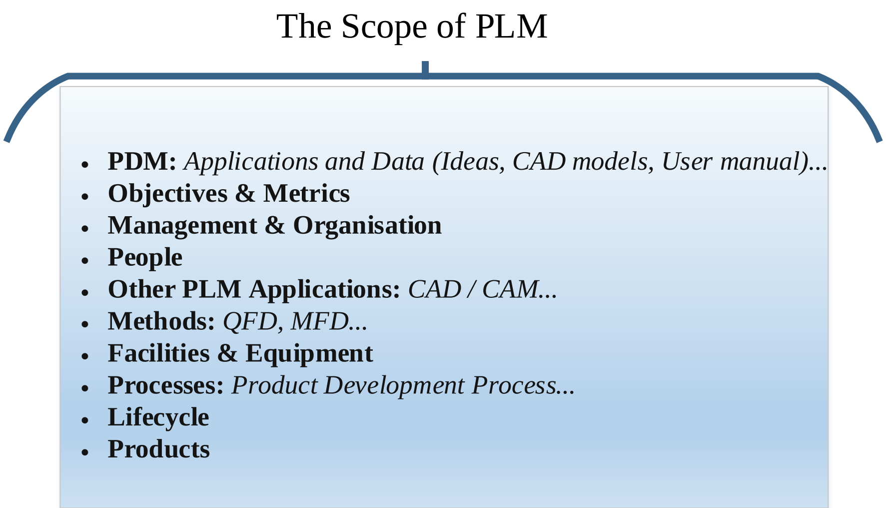


Figure 45. The Scope of PLM.

One of the most important parts of the PLM system is Product Data Management (PDM), which primary purpose is to manage all product data that is created and used throughout the product lifecycle. There is a huge amount of product data in the PLM system, which makes it difficult to manage. The PDM system is used to capture the product data when it is created and to keep all this data under control, in order to provide the right information at the right time to the users. However, it should be stated that a PDM system is not a result from the need to manage large volumes of product data generated by the PLM applications, it is actually mainly used to improve productivity and to respond more flexibly to the customer demands.

Product data model and product structures

The *product data model* is a conceptual model of the product, in which information of the product and the relations between various information objects are analysed at a general level, i.e. generic level. This information can later be used when creating all individual product variants.

A product structure is a model (related to *physical decomposition*), which represent the product information and how the information relates hierarchically to other pieces of information, see Figure 46. In other words, parts, modules, service elements and documents are attached to the product and to each other through the product structure. The product structure can also contain configuration rules in the structural relations, which makes it possible for e.g. the design function at a company to inform sales about allowed combinations of modules. Modular products that are configured according to customer specifications are always created with the help of configuration rules, which usually are handled by the PLM system.

The product structure can be based on a generic product data model, i.e. a *generic product structure* in many cases, or according to a *part list*, also referred as *BOM* (Bill of Materials) (Saaksvuori & Immonen, 2008).

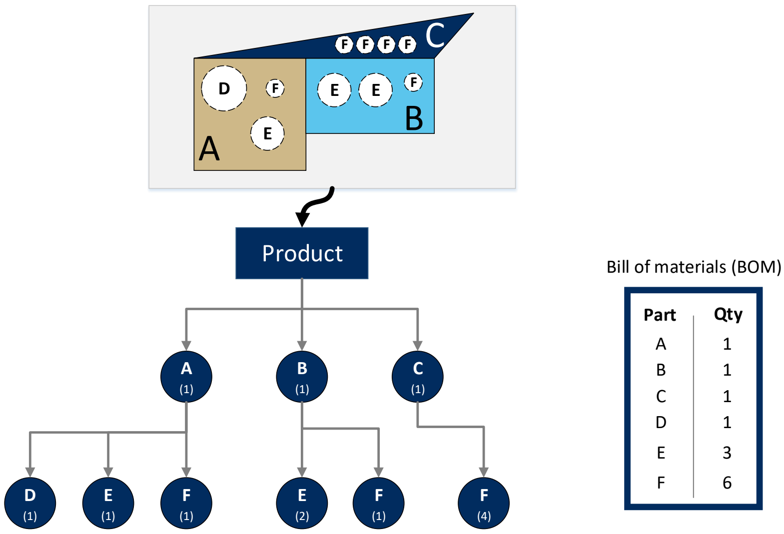


Figure 46. An example of a product structure (engineering view) and the corresponding part list (eBOM).

The methodology used to describe a product structure is usually object-oriented. An object (usually named *item*) is a data element, which describes a certain product subsystem, module, component or part. Each object can also be connected to attributes, which contains information about the properties of the object, e.g. a component object might include its weight, item number and cost.

The product structure can therefore be seen as the heart of a PLM system, where different viewpoints of the same product structure may be needed depending on function within the company, e.g. engineering, manufacturing or sales. The product structure also provides the foundation for some of the main functions of a PLM system.

A part list, or BOM, is a list of each individual component in a product, see Figure 46. Usually, the list also contains information about e.g. the part number, the source of the part, and the purchase order number (if outsourced to a supplier). There are different types of BOMs, e.g. eBOM (engineering BOM) and mBOM (manufacturing BOM), depending on the need within a company.

The difference between a *product architecture* and *product structure* should be noticed. The product architecture is the main model of a product, which brings together the product structure and the function structure, i.e. the mapping between functions and technical solutions (Brecher, 2012).

Modern PLM applications are able to handle many product structures for the same product, where each *individual product structure* represents a specific configuration of the end product, i.e. not based on a generic product structure. However, the management and updating of relations between separate product structures is a huge task, and may become impossible or very time consuming in reality for complex products. It is therefore important to consider if it is reasonable to have individual product structures, or if a generic structure shall be used.

The generic, or general, product structure is a structure developed for a product portfolio, instead of a single product variant. It contains several interchangeable and configurable components, in order to describe all possible product variants. This means that a generic product structure and the corresponding product portfolio is created during the product development process, while the individual product variants are formed during the order-delivery process. When a product variant is customised according to a customer wish, it is called product configuration or a configuration process. In this process, an individual product structure is created from the generic product data model.

A generic product structure therefore supports a modular product architecture, especially when it is possible to create many product variants from the product portfolio. However, depending on company strategy, it may sometimes be suitable to describe a modular product with individual product structures. This will mainly be beneficial if there is a high focus on the module drivers; *carry over* and *common unit* or *upgrading* and *technology push*, mainly since these drivers focus on few variants. It is therefore important to describe a product in a suitable way, depending on company strategy and product architecture. A generic product structure will also lead designers to think about alternatives to parts already used in the product. This will limit the number of alternative designs to an existing solution (Scania, same need, identical solution).

An analogy between an *individual product structure* and a *generic product structure*, is the different ways to describe all food dishes that can be made from the ingredients in a kitchen. One way is to describe the ingredients (technical solutions) to a specific dish (product variant), this will result in “individual structures” for each and every dish, which makes it hard to see the similarities, unless there are few dishes. Another way is to describe all available ingredients in the kitchen (generic product structure), which makes it a lot easier to see similarities between the dishes. Obviously, it is not possible to make one dish without first knowing which ingredients to choose from, i.e. a recipe is needed.

One of the most important functions of a PLM system is to manage and maintain all product structures. Typically, product structure management includes *version management*, *change management* and *configuration management*.

One of the most important features in a PLM system is change management, since it provides control and traceability to the change process for all products, and allows all departments within a company to get the latest and correct information. The change may affect e.g. a document, technical solution or a structure. The change process either begins with an Engineering Change Request (ECR), or directly with an Engineering Change Order (ECO). The reason for a change can for example be a design improvement, an idea for a better technical solution, or a new customer demand.

As earlier stated, a modular product architecture makes it easy to make changes, due to the one-to-one mapping between technical solutions and functions, as well as the standardised decoupled interfaces. The development of a modular product architecture, described with an generic product structure, will therefore have an extra high focus on change management.

2.9 The Scania Product Description

The product description at Scania has been continuously developed over many years to support the principles of modularisation and the company strategy. Since Scania aims at creating as many product variants as possible (customer first), which at the same time should have high performance, the product description needs to handle extremely many variants and complex configuration rules. This strategy has resulted in a product description based on generic product structures, handled by multiple systems.

The heart of the PLM system at Scania is the generic variant-coded *design structure*, which was created to describe the modularised Scania product (Truck, Bus and Engine) in an efficient way. This type of product structure is generic, since it does not describe a single product variant, but rather the entire product portfolio, i.e. the Scania Modular Toolbox.

Since the Modular Toolbox at Scania is described with a generic product structure, it means that the individual product variants are only formed during the order-delivery process, when an actual customer demand is identified. It also means that the product development process consists of development of the Modular Toolbox, i.e. not individual product variants (except S-order).

One important object type in the product structure are variant codes, where FPC codes (Functional Product Characteristic) are widely used. The variant codes therefore describe which component series and performance steps that should be used in a specific product variant to a customer. The variant codes also describe other properties, e.g. if the steering wheel should be on the left or right-hand side. The variant codes are also used as configuration codes that acts as a “key”, which allows a specific product variant to be described from the initially locked generic product structure.

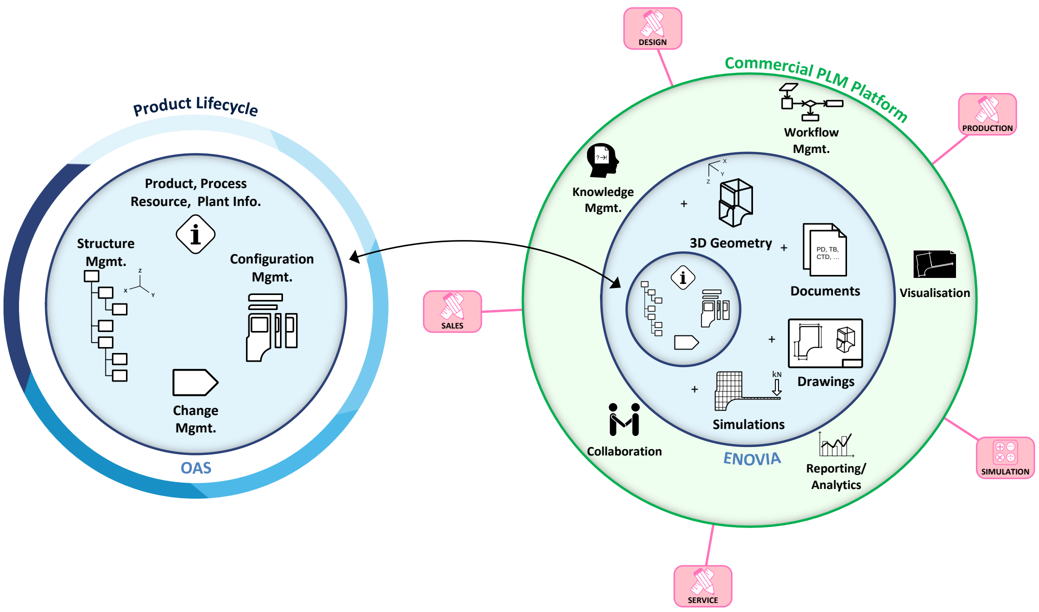


Figure 47. The interface between OAS and ENOVIA.

The Object And Structure tool (OAS) is a database platform, which is used to manage product data throughout the lifecycle. The OAS platform was developed by Scania, and includes PDM system functionality which has been developed and adapted to the requirements of the Scania modular system and change management.

However, the management of CAD data is done by the commercial software ENOVIA, where OAS only contains a reference to the stored data, see Figure 47. The OAS and ENOVIA platforms are therefore linked with an interface, which allows a specific product variant to be generated as a complete CAD model.

The management of the *assembly structure* is neither a part of the OAS system. Instead, this structure is handled by a separate Scania developed system called MONA, the synchronization of data between OAS and MONA is therefore carried out manually to a large extent. The other types of structures at Scania will be presented in a later section.

Conditions

The conditions in a product structure at Scania determine if a combination of two objects are valid or not, i.e. it enables configuration rules as well as change management. In default mode, each condition is locked (the switch is open), meaning that the objects are not allowed to be combined, see Figure 48. In the opposite way, a valid condition allows the two objects to be combined. Conditions are therefore used in many types of structures at Scania.

The configuration switch in a condition consists of one or several logic gates, in order to determine if the switch should be open or closed, i.e. not valid or valid. Logic gates are widely used in electronics to perform logical operations on one or several binary inputs. This theory has been used when designing the conditions at Scania.

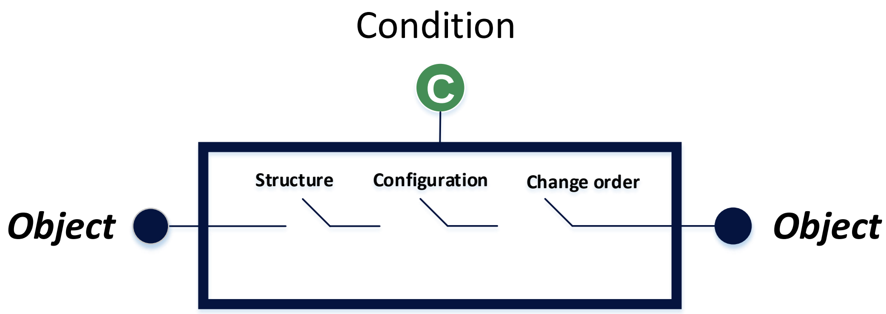


Figure 48. The design of a condition.

The logic gates in a condition is unlocked with the help of variant codes, hence the selection of variant codes during the configuration stage acts as a key to the locked generic product structure, leaving only some of its branches valid and visible. Hence, variant codes have two important functions, both to describe a property used in a configuration, and also work as a key which is able to unlock a locked condition.

The following bullets define under which circumstances a condition is valid or not.

- A condition may be valid depending on type of structure, e.g. design structure (KS).
- A condition may be valid depending on a specific combination of one or several variant codes (i.e. depending on key).
- A condition may be valid depending on time and change (ECO).

Without a key (variant code string), the structure is locked, meaning that nothing is described. As the key opens “doors” (i.e. the conditions) during the decomposition in a structure, new lower level doors will appear, see Figure 49. If a door is not opened, the lower level doors will not be shown. For example, a cab needs to be selected before it is possible to select a steering wheel.

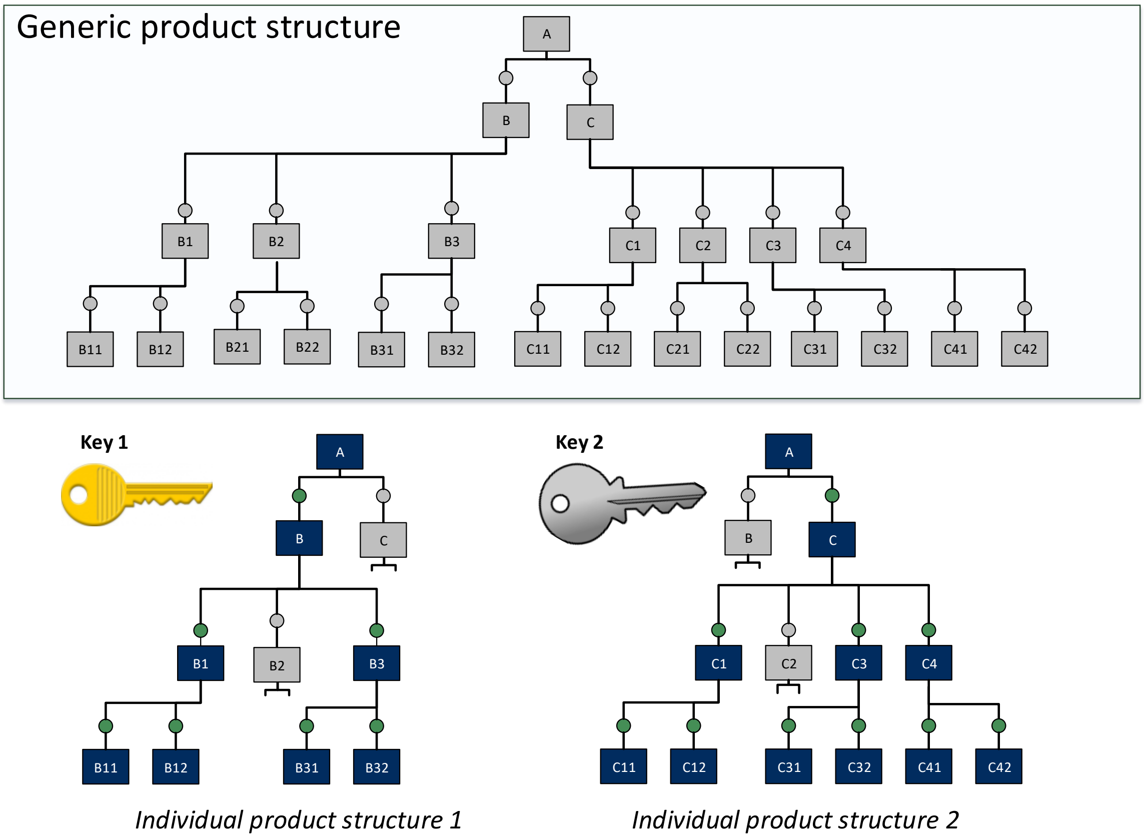


Figure 49 The unlocking process of a generic product structure.

It should be stated that conditions are not used between two objects if there are no choices, i.e. when something is always valid (not necessary to have a door to lock if everyone should have access).

The key does not know about allowed combinations, and may therefore have opened doors that are not allowed to be open in combination together. Therefore, the key needs to be checked against another structure, describing which doors that are allowed to be open together, i.e. the Variant Combination Register (VCR), see Figure 50. After this stage, the key may be modified so that it can only open an allowed combination. During this stage, new variant codes may also be added with the help of *EXIT calculations*. These types of calculations use the incoming variant codes to perform different types of mathematical operations, in order to create new codes which includes the result. An example of a calculation is the propeller shaft length, since it is highly dependent on axle distance, engine torque and type of gearbox.

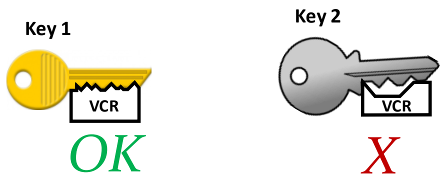


Figure 50. Illustration of a VCR check.

As seen in Figure 50, “Key 2” opened a combination which was not allowed, meaning that the product structure contained combinations that were not allowed according to the VCR.

For example, it is not allowed to combine the most powerful engine with the weakest gearbox. These restrictions are specified by the designers in the *technical specification*, before they are inserted into the VCR. Obviously, there should be as few restrictions as possible in order to create as many product variants as possible.

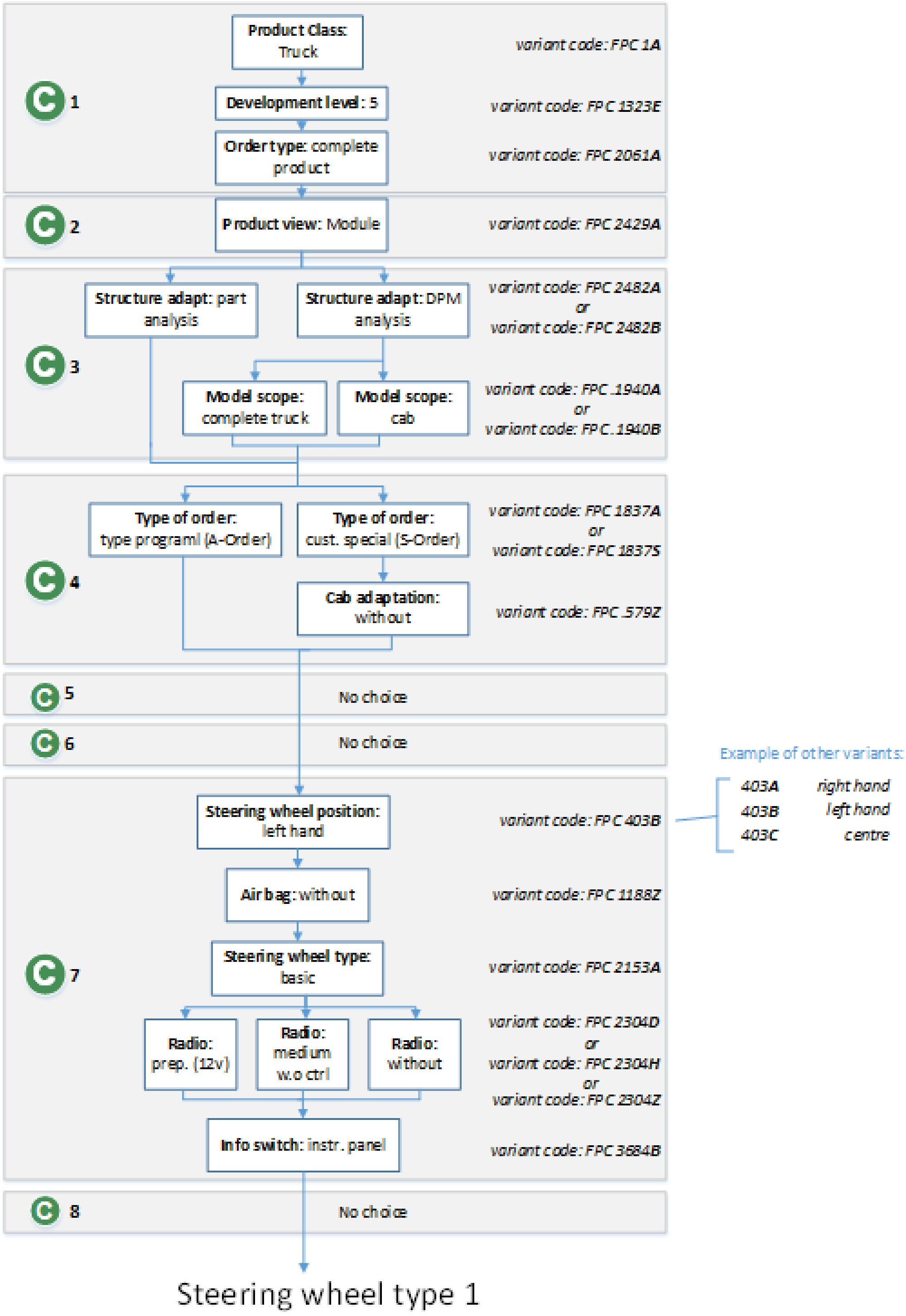


Figure 51. Example of conditions and variants codes in KS, based on Figure 52.

In reality, the VCR check is done several times before a product variant is manufactured, since the conditions are constantly being changed with ECOs. This methodology allows Scania to make changes very often, and at the same time always know if a configuration is valid or not.

An example of the unlocking process in the design structure (KS) will now be presented. If one object is the *steering wheel* and the other one is *steering wheel 1 LHD, see "Condition 7"* in Figure 51, the following variant code string (part of the key) needs to be chosen in order for condition 7 to be valid (unlocked):

FPC: *403B* ***and*** *1188Z* ***and*** *2153A* ***and (****2304D* ***or*** *2304H* ***or*** *2304Z****) and*** *3684B.*

If these variant codes are chosen, the condition will be valid, assuming that the structure is KS and that no ECO exists.

However, in order for the steering wheel type 1 to be selected, all eight conditions need to be valid, see Figure 51. Please notice that a VCR check also needs to be done before it is possible to know if the selected variant code string is valid.

The Design structure (KS)

The *module view* in the *design structure* plays an important role at Scania since it describes what the Scania product (i.e. Truck, Bus and Engine) consist of. This means that the design structure contains the Scania Modular Toolbox, even though it may be hard to see the complete picture, since the same component (or performance steps of the component), may be found at multiple locations in the structure. The branches in the structure therefore only shows a limited view of the complete range of modules (component series) and module variants (performance steps), furthermore the structure describes the interfaces in a very limited way.

There are many different object types in the design structure, which are used to describe the hierarchy level and whether the object is e.g. a technical solution or a position, see Table 8. The design structure does therefore not only contain technical solutions, it also contains spatial information (i.e. coordinates) about the location of some technical solutions. This information is mainly used to create 3D models in the CAD environment.

Table 8. The different object types in the design structure.

Object type		
EP	Enterprise Product	
PCL	Product Class	
PV	Product View	
MPU	Main Product Unit	
MU	Module Unit	
PU	Product Unit	
A	Part	
RCS	Reference Coordinate System	Spatial information
RP	Reference Position	
GP	Geometry Position	
G	Geometry Space	

Notice that the object type *part (A)* is used to describe many types of information objects, which are not only related to technical solutions. For example, documents, drawings and services are also referred as parts at Scania.

An important rule when both developing and describing a component is that it should not be dependent on the choice of another component, as long as it is possible (i.e. decoupled interfaces). It is also important to state that the design structure should not contain configuration rules, since they should only be defined in the VCR, i.e. a separate structure with another purpose.

The design structure also has another view called *functional view*, which consist of a separate structure. This structure describes use cases (UC) and scenarios (SCN), and is used to create a SOPS-file (Scania On-board Product Specification) during the production of a product variant.

One important function of the SOPS-file is to configure parameters in the different Electronic Control Units (ECU), within the electrical system of a product variant. Since there are extremely many product variants at Scania, the embedded software is parametrised, in order to eliminate the need of extremely many software variants. For example, the software in the *engine control unit* is parametrised since it is dependent on the cab configuration.

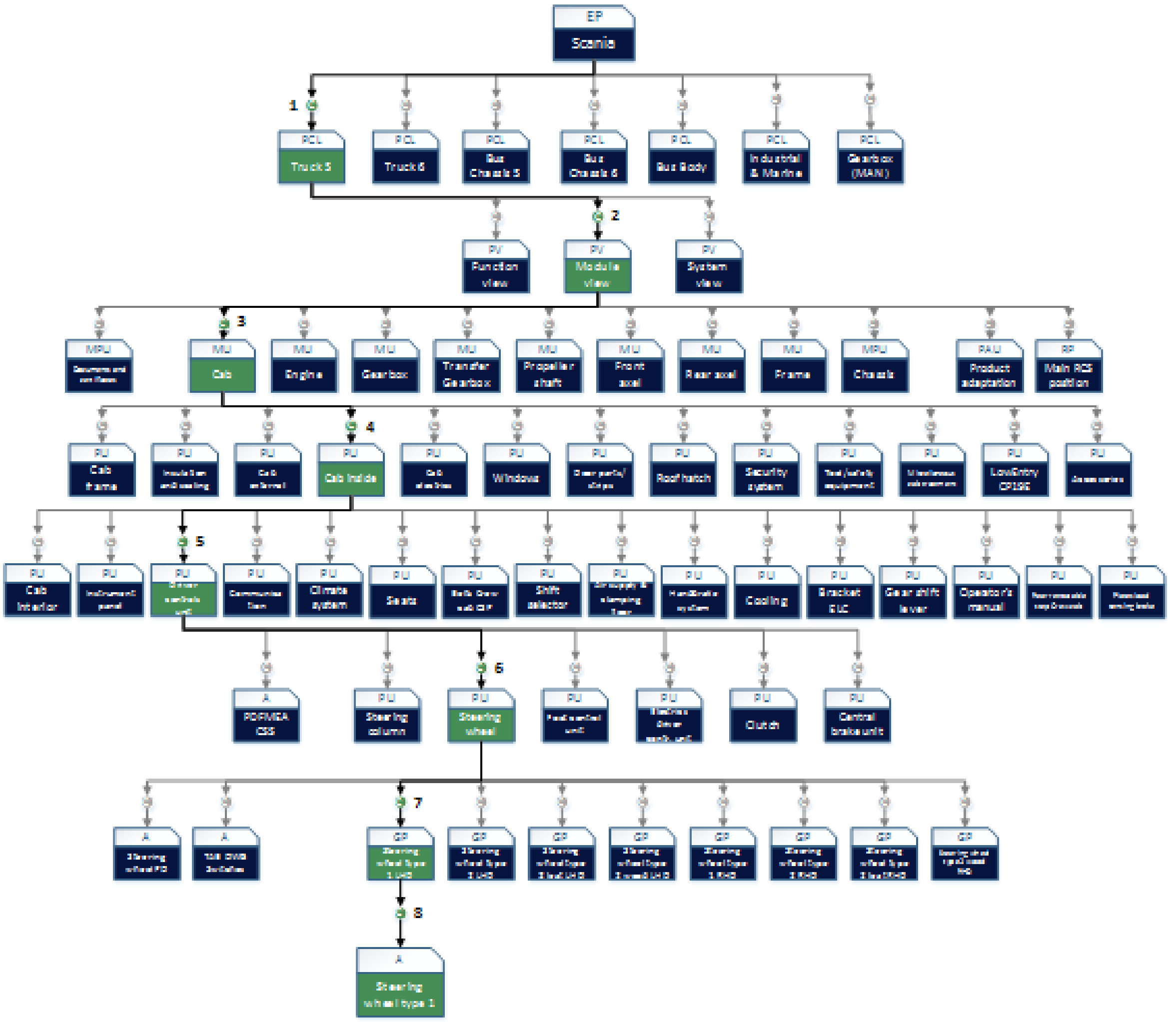


Figure 52. An example of a KS that shows how a steering wheel is described

The TCR and VCR structure

The Translation Code Register (TCR) at Scania consist of a structure (with conditions as earlier described), which is used to *translate* a simplified incoming variant code string from the sales department, to a more detailed variant code string, which is needed at other departments at Scania.

An analogy to the function of the TCR will now be presented. If a customer wants pancakes, the chef does not ask the customer what ingredients he should choose from in the kitchen (i.e. generic product structure). The chef may however ask some carefully chosen questions, e.g. do you want blueberry to the pancakes? Clearly, there must be some kind of translation between the customer and the chef, in order to simplify and speed up the configuration stage, which is the TCR at Scania.

The Variant Combination Register (VCR), is a method (also containing a structure) used to describe allowed combinations of objects, and add variant codes (*EXIT calculations*) which are missing from the TCR. This means that even if a product variant has been configured according to a customer specification, it may not be valid due to the selected combinations.

A variant code string is checked against the VCR structure in order to verify that there are no combinations which are not allowed, i.e. checking if the doors the key can open are allowed to be open at the same time. For example, it is only possible to select an air bag (FPC 1188) to the steering wheel, if a seat belt pre-tensioner and an adjustable steering wheel have been chosen. Another example is the allowed combination between engines and gearboxes, see Figure 53. These combination restrictions are then inserted into the VCR in the form of a structure.

	Mechanical Gearboxes						
Engine Type	GR875/R	GRS895/R	GR905/R	GRS905/R	GRSO905/R	GRSO925/R	GRSO935/R
DC13 102		X	X	X	X	X	X
DC13 112		X	X	X	X	X	X
DC13 121		X	X	X	X	X	X
DC13 124		X	X	X	X	X	X
DC13 159		X	X	X	X	X	X
DC13 147		X	X	X	X	X	X
DC13 106		X	X	X	X	X	X
DC13 101			X	X	X	X	X
DC13 111			X	X	X	X	X
DC13 125				X	X	X	X
DC16 04				X	X	X	X
DC16 06				X	X	X	X
DC16 19				X	X	X	X
DC16 22				X	X	X	X
DC16 101				X	X	X	X
DC16 05				X	X	X	X
DC16 18				X	X	X	X
DC16 03				X	X	X	X
DC16 104				X	X	X	X
DC16 102					X	X	X
DC16 08					X	X	X
DC16 17					X	X	X
DC16 21						X	X
DC16 103						X	X

Figure 53. Allowed combinations between some engines and gearboxes.

It should be noticed that the VCR includes restrictions which are purely created due to marketing and branding concerns, i.e. it has nothing to do with if the combination actually works or not in terms of technical aspects. An example of this is that only some cab interiors are valid if the V8 Engine has been chosen, e.g. it is not possible to select a V8 Engine and the “basic” interior.

Since there are many complex combination restrictions in the Scania Modular Toolbox, it is important that all restrictions are defined in the VCR structure. In this way, it is always possible to verify if a configuration is valid or not.

The check in VCR is performed by unlocking the conditions in the structure, with the same key (variant code string) as used to unlock the design structure (KS), i.e. the exact same methodology. However, the result of this unlocking process will yield if the selected combinations are valid or not, i.e. if the key is valid or not, see Figure 54.

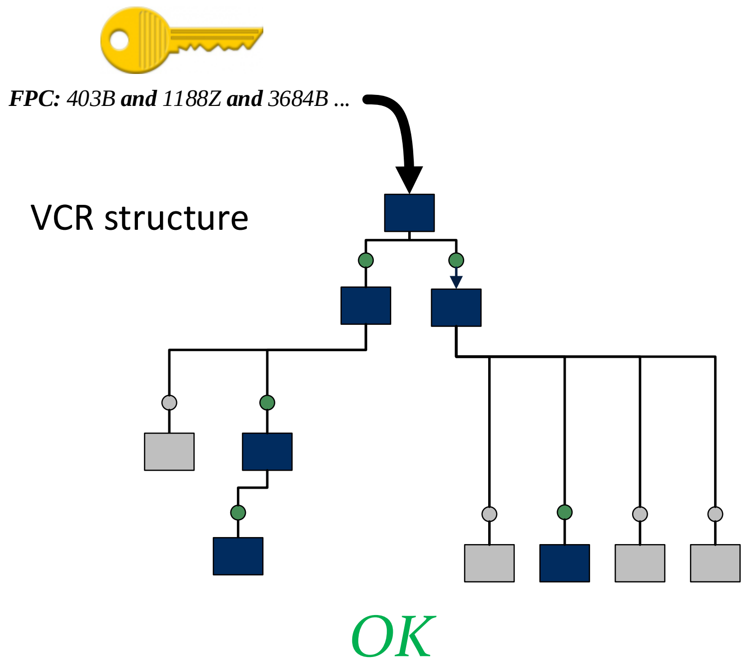


Figure 54. Illustration of the VCR check.

The order-delivery process

One of the most important processes which the PLM system needs to handle at Scania, is the *order to delivery process*. The main reason why this process is important, is because all product variants which are to be sold and manufactured, needs to go through this process.

The process starts with a configuration stage, handled by the system ETEL and in combination with the Scania Vehicle Optimiser (VO). ETEL also contains a limited version of the VCR, which makes it possible to see combinations that are allowed (in an early phase). The input information to the configuration is mainly the operational factors, as well as an earlier Scania variant code string (if available), in order to compare the new configuration with the current old product. After the configuration, the order is checked and the logistics and production planning begins. It should be stated that the customer offering will be dependent on the market where the product is to be sold, this means that some remote sales location (e.g. Thailand), may pack the variant codes to limit the number of configuration choices.

After completing the configuration and order check stage, the description initially takes the form of the so-called C-specification, which consists of an incomplete variant code string, i.e. the key is not yet complete.

As earlier described, the *translation* of a simplified variant code string to a more detailed variant code string, is performed automatically by the TCR. After the translation, the VCR is used to check the specification against the allowed combinations, as well as adding variant codes. The output of this stage will be a so-called V-specification, which means that the variant code string is verified and complete (containing about 900 -1500 variant codes) and ready to be used when manufacturing a specific product variant, see Figure 55. After this stage, the variant code string is split into different production units at Scania, depending on where the technical solutions should be manufactured.

The SOPS-file (Scania On-board Product Specification) is used to configure parameters in the different Electronic Control Units (ECU), within the electrical system of a product variant.

The SOPS-file is generated from the V-specification in SIDE (Scania Individual Database for End of line programming).

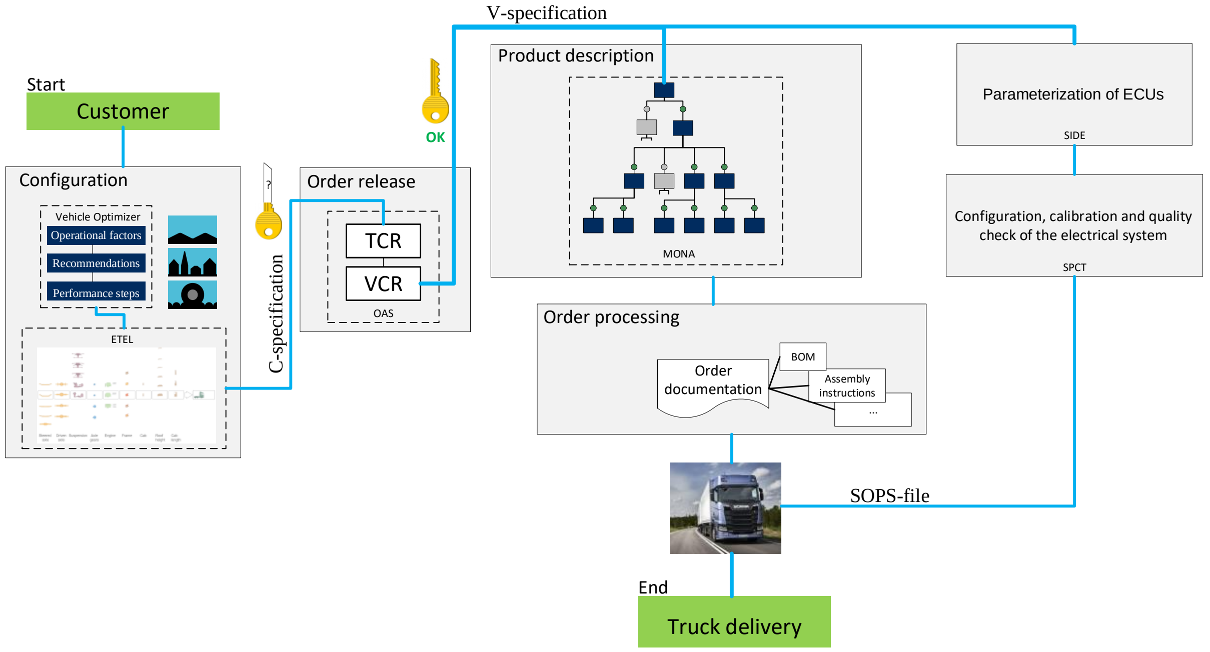


Figure 55. Simplified view of the order-delivery process at Scania.

The SPCT (Scania Production-Computer Tools) consists of multiple computer systems, which are used during production in order to configure, calibrate and check the quality of the electrical system of a product variant.

Different product structures at Scania

The *design structure (KS), the VCR and TCR structure* have been described in earlier sections. However, there are many other structures at Scania which also are important for different users. The common thread in most product structures at Scania is the variant code string, i.e. the key, which is used to unlock a product structure, in order to describe something specific (i.e. a product variant) from the generic description. The variant code string therefore aims at creating a standardised interface between the different structures.

The *assembly structure* is managed by the Scania developed MONA system, which also consists of objects with conditions in between. The objects in this structure describe how a product variant shall be assembled, e.g. assembly area and task etc., which obviously makes the structure look different from the design structure. Due to this reason, the conditions between the objects in the design structure (KS) cannot be translated in a simple way to the assembly structure, this work is therefore done manually to a large extent.

After a product variant has been manufactured, the spare part kits and service information is stored in the MULTI system, where it is possible to get information related to a specific product variant.

The *prediction variant structure (PVS)* is located in OAS, and is used to predict the number of product parts (part numbers) which will be needed at the different production units, by estimating the number of product variants which will be needed in the future. The estimation is based on both history, statistics and actual orders.

The *rebuild structure (RBS)* contains information about which reconstructions that have been performed by Scania, as well as what parts that needs to be changed from the original configuration. The RBS structure also contains the original conditions, including the variant codes and ECOs. This makes it possible to describe a rebuilt product variant, by reversing the original SOPS-file in the VCR, followed by changing the variant codes in the C-specification, before creating a new SOPS-file.

Product structure usage

The product structures, as well as the variant code strings, are used in many applications and processes at Scania. Some of the applications which are mainly used during R&D, will be presented in this section.

The *design structure* is used to perform different types of *geometry analyses* with the help of DMU (Digital Mock-Up) in the CAD environment. This is a very important feature at Scania, since there are extremely many product variants which needs to be verified. The geometry analysis mainly involves clash and tool accessibility analyses. The difficult part with these types of analyses is to exclude product variants which are not valid due to combination restrictions in the VCR structure.

A very important property of a vehicle is its *mass*, including the *centre of mass*. However, since all object attributes in the *design structure* do not contain mass information, it is not possible to simply add all masses in order to get the total. Therefore, the Weight Calculation Program (WCP) is used at Scania to estimate the mass, as well as the centre of mass, by the help of variant code strings.

The design of the *truck frame* is highly dependent on the configuration, therefore each frame is individually designed (parametrised) in terms of length, strength, colour, thickness and hole punching etc. These parameters are generated dependent on the incoming variant code string, i.e. depending on how the product variant is configured. Another component which also is highly dependent on the configuration is the *propeller shaft*, since its length is dependent on axle distance, engine torque and type of gearbox. Except from these examples, there are *many other components* which also are dependent on configuration, e.g. cable harness etc. The dependent parameters are optimized for each product variant configuration, with the help of EXIT calculations in the VCR or TCR structure. As earlier described, the configuration of the parameterised *embedded software* (configuration dependent) is performed with the help of the SOPS-file, which is generated from the V-specification by SIDE.

3 CASE STUDY

This chapter contains the implementation stage of the research project. During this stage, specific main Scania components were first identified and selected, in order to exemplify the difficulties when modularising and describing a multidisciplinary product. After that, semi-structured interviews were performed at Scania, in order to get an insight of how the modularisation process and product description is actually done. Finally, an extensive analysis of the selected main components concerning modularisation and product description was performed.

3.1 Identifying main components

During the background study, it was identified to be especially interesting to investigate a subsystem that covered different technical disciplines, in order to exemplify the difficulties when modularising and describing a multidisciplinary product with complex relations between the components. When selecting the main components to investigate, one criterion was therefore that it should contain *hardware, electrical system* and *embedded software.*

A main component that fulfils these criterions is the *gearbox*, and since Scania develop this subsystem by themselves, they have control over the product architecture and the complete design. The gearbox was therefore identified to be a good representation of how Scania has modularised and described their product. The gearbox is only a part of the complete powertrain, see Figure 56, but plays an important and central role for the vehicle propulsion.

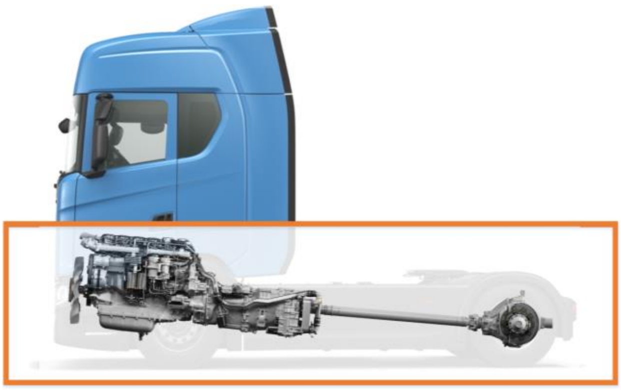


Figure 56. Cutaway illustration of a Scania powertrain in a truck.

The powertrain normally includes engine, gearbox, propeller shaft, axle gear, half shafts and hub reduction gears, see Figure 57. Together, all these components contribute to the propulsion of the vehicle.

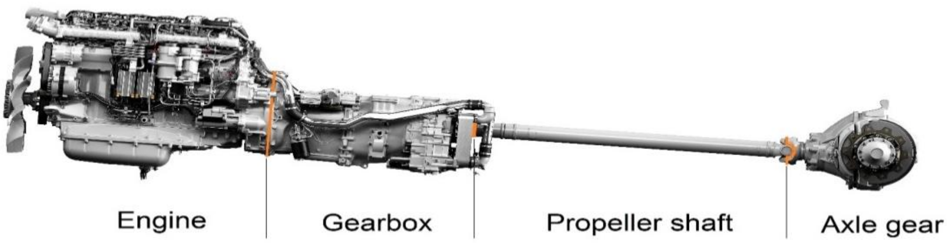


Figure 57. Illustration of a Scania powertrain.

Since there are many variants of the gearbox, a delimitation needed to be made due to time limitations. Only one variant which is delivered to MAN (GRSO905R), see Figure 58, was therefore selected to be investigated, since this variant highlight some of the difficulties by Scania also being a first-tier supplier.

In addition to the gearbox, the propeller shaft was also selected to be investigated, mainly since it is not manufactured by Scania and is parametrised dependent on the overall vehicle configuration. This reveals other important aspects of the modularisation and product description at Scania.

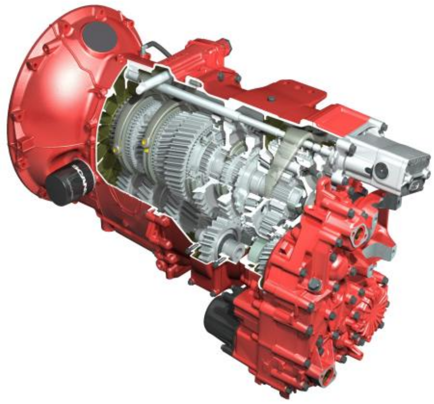


Figure 58. Cutaway illustration of a Scania GRSO905R gearbox.

3.2 Interviews at Scania

The observational part of the case study was performed with semi-structured interviews (Kvale, 1997), to get an insight of the modularisation process and product description, from the perception of several designers at Scania. The interviews consisted of a set of semi-structured questions, in order to guide the conversation in a partly predefined direction, with one respondent at a time. Each respondent was asked to speak openly about one question at a time, according to their own experience of how they work and think in reality, i.e. not according to how things are supposed to be according to the Scania principles or methods.

This interview approach was chosen in order to allow the respondents own words and experience to shine through. This gave the respondents some freedom to talk more about what was of importance to them, and also made it possible to acquire information which was not thought of in advance.

The entire case study was performed with "The seven stages of the qualitative research interview process" as a guideline, see Figure 59.

The seven stages of the qualitative research interview process

1. Defining the purpose and topic.
2. Planning.
3. Interview.
4. Translation.
5. Analysis.
6. Verification.
7. Reporting.

Figure 59. The seven stages of the qualitative research interview process (Kvale, 1997).

All respondents which were interviewed during the case study, worked with gearbox development at Scania. In order to investigate if there were any difference between the different technical disciplines, engineers within *hardware*, *electrical system* and *embedded software design* were interviewed. In addition to that, both senior and junior engineers were interviewed at each technical discipline, which resulted in a total amount of 6 respondents. This was done in order to study if the amount of experience affected the answers in any way.

The selection of respondents was performed with the assistance of engineers working with the overall gearbox design and configuration. It should be stated that 2 of the 8 persons which were asked if they wanted to participate in the case study, declined the offer, since they felt too inexperienced and claimed to have little experience in the area of modularisation and product description.

The following respondents were interviewed with the semi-structured question guide in Appendix A:

Respondent	**At Scania**	**Role**
R1	9 years	*Senior Mechanical Engineer,* with extensive experience in mechanical design of the gearbox
R2	1.5 years	*Junior Software Engineer* with limited experience in embedded gear control software design
R3	15 years	*Senior Software Engineer* with extensive experience in embedded gear control software design
R4	2 years	*Junior Electrical Engineer* with limited experience in the electrical cable harness design of the gearbox
R5	3 years	*Junior Mechanical Engineer* with limited experience in mechanical design of the gearbox
R6	10 years	*Senior Electrical Engineer* with extensive experience in the electrical design of the gearbox

As seen in the interview guide, the discussion with the respondents focused on *Nomenclature* and terminology, *Modularisation principles* and *Product description*.

3.3 Functional analysis

In order to understand the identified and selected gearbox (GRSO905R), including how the technical solutions were linked to the functions, a functional decomposition was performed by using a *function-means tree*, see Figure 60. It should be stated that this analysis was done by the author and with assistance of senior expert engineers within the gearbox department, since this type of information did not exist at Scania.

This analysis did not include the propeller shaft, since it is a separate component with one main function, to transfer mechanical energy.

The main function of the gearbox is to control the angular speed and torque from the engine to the wheels, when operating during a variety of load and road conditions. Due to the torque vs. speed characteristics of a normal combustion engine, the engine cannot achieve this function by itself, hence a gearbox is required.

When decomposing the main function of the gearbox, multiple sub-functions were identified. It should be stated that all these sub-functions are needed in order to enable the main function. Many of the functions were realised with multiple components, especially the shift torque function (i.e. the function used when shifting gears). The technical solutions in the function-means tree were investigated and selected by looking in the design structure (KS) at Scania, see Figure 63.

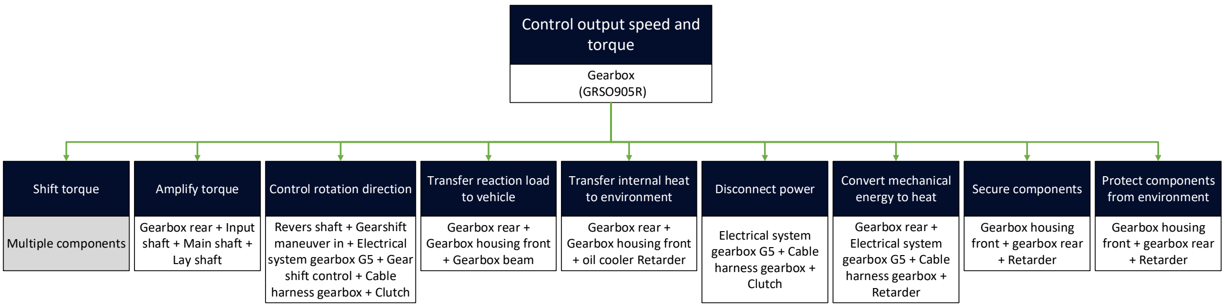


Figure 60. Function-means tree of the selected gearbox.

Based on the result of the function-means tree, a corresponding dependency matrix was created in order to illustrate the degree of modularity of the gearbox, see Figure 61.

	Gearbox rear	Gearbox housing front	Input shaft	Main shaft	Lay shaft	Reverse shaft	Gearshift maneuver in	Oil cooler	Electrical system gearbox G5	Gear shift control	Cable harness gearbox	Gearbox beam	Clutch	Retarder
Shift torque	●	●		●	●		●		●	●	●		●	
Amplify torque	●		●	●	●									
Control rotation direction						●	●		●	●	●		●	
Transfer reaction load to vehicle	●	●										●		
Transfer internal heat to environment	●	●						●						●
Disconnect power									●		●		●	
Convert mechanical energy to heat	●								●		●			●
Secure components	●	●												●
Protect components from environment	●	●												●

Figure 61. Mapping of functions and technical solutions.

Some of the gearbox functions rely on technical solutions which are not part of the gearbox. For example, the oil cooler needs to be connected to an external cooling system, which is part of the engine. This creates a dependency between the gearbox and engine in terms of cooling. In order to increase the degree of modularity, the gearbox would need a separate cooling system, e.g. a radiator with a fan mounted on the outside of the gearbox housing. Obviously, this would result in lower performance in terms of weight, volume and cost, but the flexibility would be increased.

After creating the dependency matrix, it was possible to estimate how coupled the gearbox design was based on the number of dependencies, see the highlighted area in Figure 62. It was clear that the gearbox design is relatively coupled, since many of the functions are realised with multiple technical solutions, especially the shift torque function. Please notice that the gearbox may have a higher degree of modularity at a higher level in the product structure, i.e. the gearbox may be a very independent module in the complete vehicle.

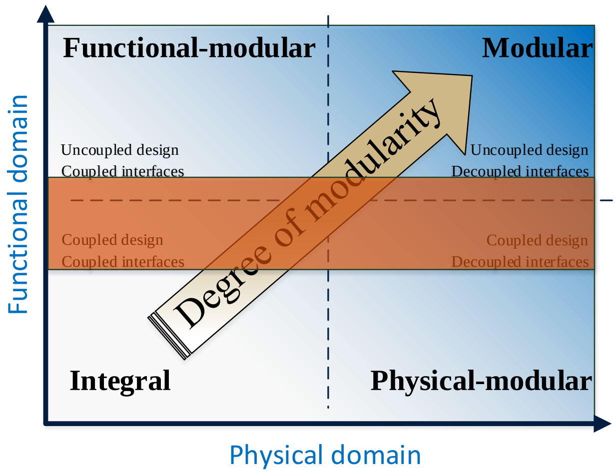


Figure 62. The degree of coupling in the gearbox design.

The means, i.e. the technical solutions were investigated and selected by looking in the design structure (KS) at Scania, see Figure 63. In order to limit the number of components to a reasonable level, due to the time limitations of the project, all screws, O-rings, gaskets and other small parts, were not considered when decomposing the technical solutions to a lower level. After talking to experts within the gearbox development, it was clear that the design structure (KS) was highly based on how the gearbox is assembled during manufacturing. This approach made the analysis harder, since it was difficult to see the complete picture of the technical solutions from a design perspective.

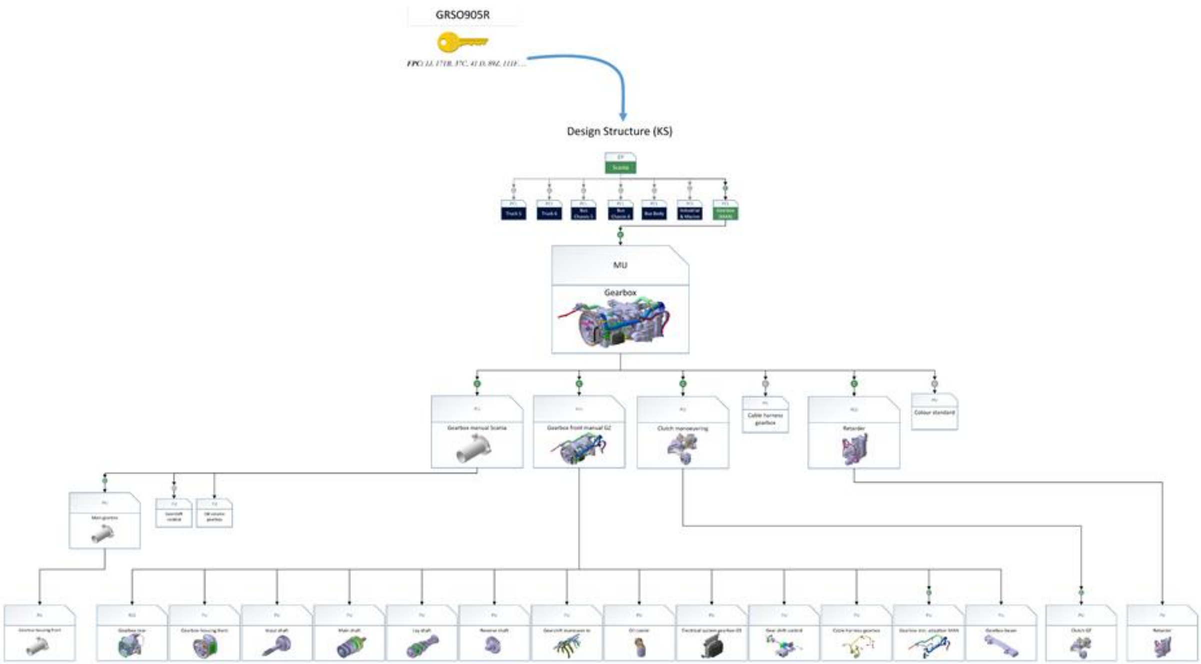


Figure 63. Physical decomposition base on the design structure (KS) at Scania.

After investigating the elements in the design structure, the following 92 components were identified, see Table 9.

Table 9. Identified components during the physical decomposition.

№	Component name	№	Component name	№	Component name
1	Sleeve	32	Noise shield PTO	63	Gear shift fork 2-3/split
2	Gearbox housing rear S9	33	Input shaft	64	Oil cooler
3	Cover reverse gear sensor	34	Input shaft gear	65	Electrical system gearbox G5
4	Bearing housing retarder	35	Main shaft	66	Gearshift housing
5	Cover PTO range	36	Main shaft gear split	67	CYLINDER
6	Sun wheel	37	Main shaft gear	68	Spring housing
7	Driver	38	Main shaft gear 2nd	69	Solenoid valve assy opti
8	Coupling sleeve	39	Main shaft gear crawler	70	Shaft manuvering OPC
9	Coupling disc	40	Main shaft gear reverse	71	Rotation speed sensor
10	Shift sleeve EDT	41	Driver	72	Lever inner gearshift
11	Coupling cone high	42	Driver	73	Cylinder side stroke OPC assy
12	Cylinder range-split retarder	43	Driver	74	Cable harness gearbox
13	Coupling cone low	44	Driver	75	MAN Water system GZ
14	Gear selector shaft split	45	Coupling disc ED	76	Bracket battery cable
15	Gear selector shaft range	46	Coupling disc ED	77	Oil cooling installation MAN
16	Gear shift fork range	47	Coupling disc ED	78	Gearbox beam
17	Gearbox flange crosstooth	48	Coupling disc ED	79	Release bearing assy
18	Output shaft	49	Shift sleeve EDT	80	Lever assy
19	Ring gear	50	Shift sleeve EDT	81	Bracket servo
20	Air pipe	51	Lay shaft	82	ECA 1.5 assy
21	Cable retainer right	52	Lay shaft gear input	83	Filter cover
22	Solenoid valve	53	Lay shaft gear 2nd	84	Oil Cooler
23	Road speed sensor	54	Lay shaft gear split	85	Valve Assy
24	Gearbox housing	55	Reverse shaft	86	Coolant System
25	Cover lower	56	Gear shift shaft 2-3	87	Shaft retarder
26	Plate	57	Gear shift shaft 1	88	Accumulator assy
27	Oil collector assy	58	Gear shift fork 2-3	89	Oil pan cover
28	Oil pump assy	59	Gear shift fork 1	90	Valve housing
29	Layshaft brake	60	Gear shift shaft R	91	Retarder housing
30	Noise shield assy	61	Gear shift fork R	92	Stator and rotor
31	Noise shield cpl	62	Gear shift shaft split		

A *component architecture diagram* (CAD), see Figure 64, was then created in order to investigate how the 92 identified components interacted with each other. The interactions were described with the relations explained in the Frame of Reference chapter, i.e. geometry, signal, energy and material transfer. The layout of the components in the component architecture diagram is not fully representing the actual physical layout of the gearbox, since the goal was to represent the relations. However, it gives a principle understanding of the gearbox design.

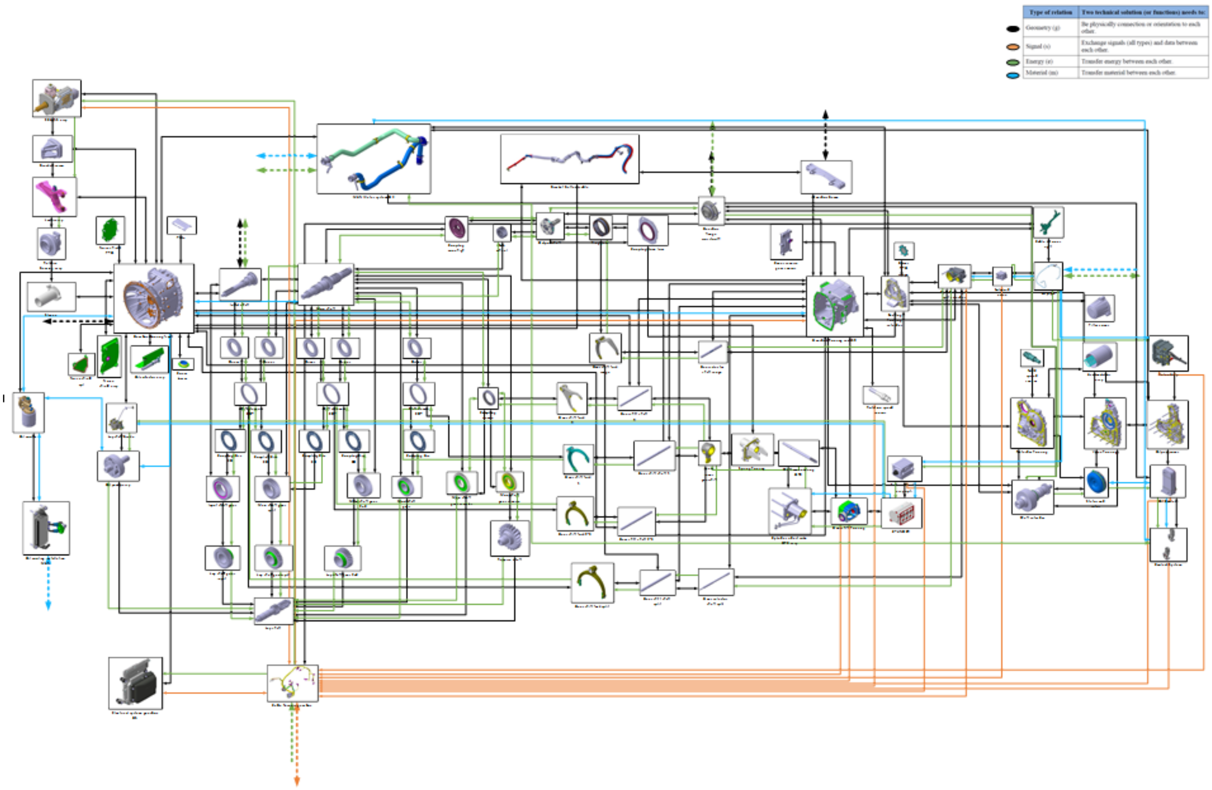


Figure 64. Component architecture diagram (CAD) of the GRSO905R gearbox.

It should be stated that the component architecture diagram was first created by the author as a means for visualisation and communication, and was thereafter analysed and verified by expert engineers within the gearbox department.

The main external interfaces to the gearbox were also visualised in the component architecture diagram. These interfaces involve oil flow from/to the oil cooling heat exchanger, mechanical energy transfer from/to the input shaft and gearbox flange cross tooth, cooling water flow from/to the retarder and finally multiple spatial relations.

The Propeller shaft

The propeller shaft was also analysed in addition to the gearbox. However, since it is a completely separate component, it was not analysed as a part of the gearbox.

In order to illustrate how the propeller shaft was designed from a modularisation perspective, an example will now be presented. In this example, a single propeller shaft between the gearbox and the rear axle gear will only be analysed, i.e. no intermediate shafts since the main principle is the same.

Depending on vehicle configuration, the propeller shaft length L_{tot} will not be constant and therefore needs to be customised for each vehicle. The *EXIT calculation* (see the Frame of Reference chapter) is therefore used in order to calculate the propeller shaft length L_{tot}, as well as the joint angles, see Figure 65.

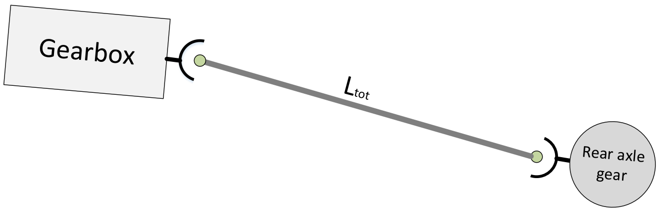


Figure 65. Side view of a propeller shaft in a truck.

The EXIT program relies on input data from multiple other components in the vehicle, e.g. wheel configuration, see Table 10.

Table 10. Input data from other components.

Transmission	Axles
Gearbox coordinates for output shaft	Axle gear coordinates for output shaft/pinion
End yoke coordinates	
Chassis	**Propeller shafts**
Wheel configuration	Strength class
Suspension type	Propeller shaft/intermediate propeller shaft
Suspension height coordinates	
Support bearing bracket coordinates	
Axle distance	

It is feasible to have different lengths of the propeller shafts since it is possible to efficiently design and manufacture a large number of variants. This is however not the case for most other components, which therefore needs to be combined in various ways in order to create the different product variants. In order to efficiently manufacture the different lengths of the propeller shaft, the tube section of the shaft is cut to the desired length, followed by a friction weld process where the two shaft pieces are integrated into one piece.

The cut length L_{kap} is calculated based on the previously calculated total length L_{tot}, the cut length is then matched with its corresponding variant code, which is added to the complete variant code string in order to cut the metal tube to the right length. The additional material F, which is consumed during the friction weld process, also needs to be considered in order to manufacture a propeller shaft with the right length, see Figure 66. In order to keep the number of propeller shaft variants down, the lengths are divided into 10 mm steps.

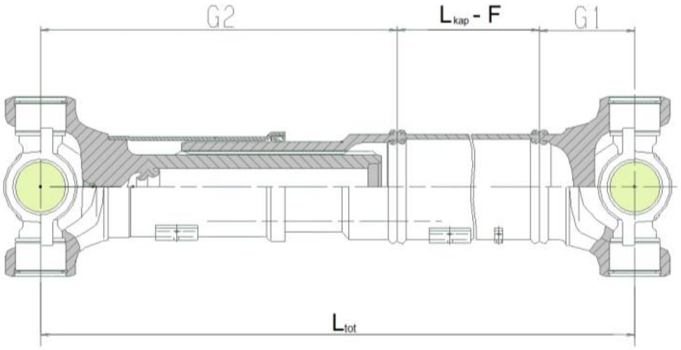


Figure 66. The propeller shaft design.

The propeller shaft example shows some of the important aspects to consider if a modular product architecture should be successfully realised. At some point, all different modules must be integrated into one system, which sometimes makes it necessary to have a module or component with a very high flexibility, which is able to “take up the slack”. Another technical solution which also needs to have a very high flexibility is most cable harnesses. A cable harness therefore has many similarities to the propeller shafts in terms of how it is parametrised and manufactured for a specific vehicle.

3.4 The Modular Architecture

After investigating if the gearbox design was coupled or uncoupled, the interfaces between the modules were also investigated in terms of coupling. As stated in the Frame of Reference chapter, a fully modular product architecture should have decoupled interfaces. This means that it should be possible to make a change in one technical solution, without requiring a change in another technical solution, in order for the overall product to work correctly.

During this investigation, the *product architecture cluster diagram* (PACD) was used to visualise the relations and thereby the interfaces between the modules. These interfaces were then analysed in terms of how coupled or uncoupled they were. The modules were also visualised in the component architecture diagram as blue shapes, see Figure 67. These modules were based on the physical decomposition defined in the design structure (KS), i.e. according to the earlier defined decomposition in the Functional analysis chapter. When identifying the modules, the level of decomposition was the same as in the dependency matrix, to enable a comparison of the level of functional and physical coupling.

As seen in Figure 67, there is a large number of spatial interfaces within the gearbox design, since many components require perfect alignment in order to work correctly. For example, the different gear pairs shares a complex spatial interface (and energy transfer interface) and therefore needs to be at a precise location. This means that a change in one component will result in a change in one or multiple other components, i.e. the interface is coupled. In addition to that, the components are fairly tightly packed (geometric nesting) in the gearbox, which also adds multiple coupled interfaces.

Despite from all spatial interfaces, there are also multiple other interfaces with energy and material transfer. For example, the gearbox ECU (Electrical system gearbox G5) needs to be mounted at a certain position on the outside of the gearbox housing, due to heat constrains of the electronics. Most of the signal interfaces are however relatively decoupled.

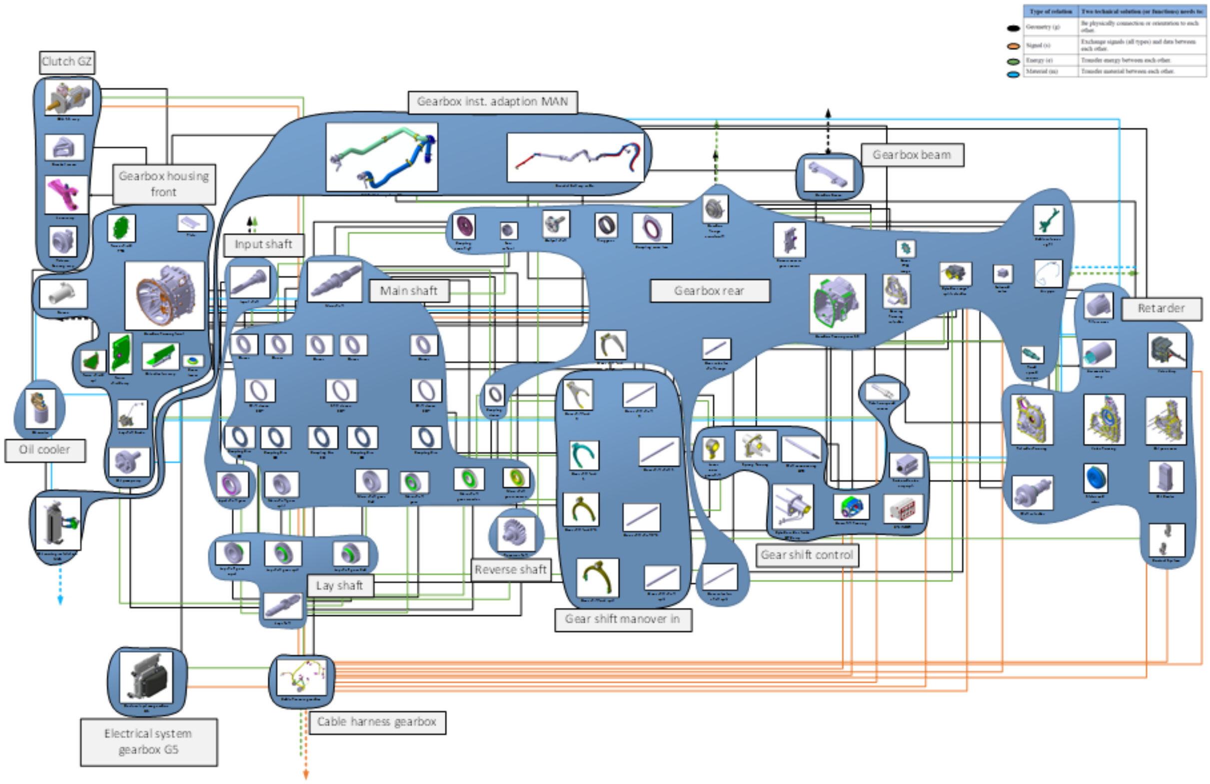


Figure 67. The modular gearbox architecture represented as a product architecture cluster diagram PACD.

It was therefore possible to conclude that the interfaces in the gearbox are relatively coupled, see Figure 68. When plotting how coupled the interfaces and function were simultaneously, it is possible to see in which area the gearbox is located in, i.e. the degree of modularity. As seen in Figure 68, the degree of modularity is relatively low for the gearbox architecture, meaning that the gearbox has a more integral than modular design. However, there are still some parts which are less coupled, allowing some flexibility in order to create different product variants. These variants, thus, needs to be defined with configuration rules (conditions), since the technical solutions cannot be combined arbitrarily.

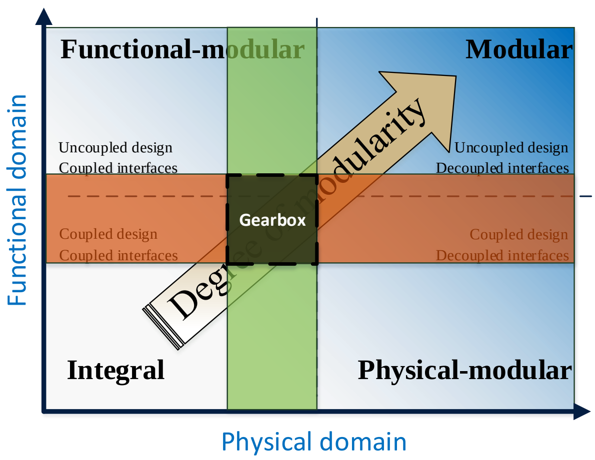


Figure 68. The degree of functional and physical coupling in the gearbox design.

4 ANALYSIS

The collaboration between Scania, Volkswagen and MAN highlights how the product is architected and described at Scania. History and business strategies have a huge impact on how modularisation principles should be applied and how the product should preferably be architected. To enable future collaboration it is of vital importance to be clear on these principles and on the terminology used in the collaboration organisations.

4.1 The interviews at Scania

This chapter contains an analysis of the six interviews performed at Scania.

Nomenclature and terminology

Most designers talk about modules during the everyday work at Scania, even if the word *module* is not officially defined. As a possible outcome of the undefined definition of a module, all designers had different definitions of what a module is. All respondents except one (R5), explained that they use the word module during the everyday work and believed that Scania has modules. However, the word *component series* was only used by one of the respondents (R1), who claimed that a component series is not the same thing as a module. This clearly indicates that most designers talk about modules (and not component series) in their everyday work, even though the word module is not defined at Scania. One respondent (R3), a senior designer within embedded software design, explained that a software module should perform some kind of function in an independent way, i.e. similar to the academic definition of modularity.

Another word which is used at Scania, without being officially defined, is the word *function*. Most respondents (4 of 6) said that they use the word function, however, all designers had their own definitions of what a function is. In the field of mechanical engineering, the word *product property* is sometimes thought to be the same thing as function, which is an unfortunate misconception. Despite the mixed answers regarding nomenclature, only a small part of the respondents believed that the nomenclature needs to be defined in a more clear and consistent way, indicating that this problem may not be very obvious for most designers. Despite the mixed answers regarding nomenclature, only one of the respondents (R3) believed that the nomenclature is not clearly defined and that miscommunications do occur due to this reason. This indicates that there is a somewhat undefined nomenclature at Scania, which may not be very obvious to most designers (especially not for junior designers).

The word *product architecture* is not commonly used at Scania. As earlier defined, a product architecture is the main model of a product, which brings together the product structure and the function structure, i.e. the mapping between functions and technical solutions. Hence, a product structure is not the same thing as a product architecture.

Modularisation

Most respondents at Scania explained that the drawback with modularisation is mainly reduced performance in some way, e.g. higher mass, higher fuel consumption etc. The cause of the reduced performance was believed to be that the complete design becomes a large compromise when everything should fit and work with everything. These answers are consistent with the definition in the Frame of Reference chapter, which indicates that the designers are aware of the drawbacks, even if they may not be mentioned during the everyday work.

One very evident result from the interviews with the different designers at Scania, was the confusion between *standardisation* and *modularisation*. Most of the respondents believed that modularisation is not the same thing as standardisation, even if they could not explain the difference. As explained in the Frame of Reference chapter, standardisation is a completely different mind-set from modularisation in terms of product variety and development process. Most of the respondents (4 of 6) believed that modularisation is not the same thing as standardisation, even if they could not explain the difference.

Another clear result was that all designers believed that the opposite to a modular product architecture is something that is specific (i.e. no variants), developed for only one purpose. This is partially right, but as explained in the Frame of Reference chapter, the opposite to a modular product architecture is an integral product architecture, which none of the respondents mentioned.

Furthermore, none of the respondents did see any problem for a function to be realised in several components or modules, since that is how the gearbox is designed today. This indicates that the designers do not think about functional independence, i.e. one-to-one mapping between function and technical solution, which is a key feature in a modular product architecture. It should be noticed that standardised interfaces do not automatically create functional independence. Respondents (R2 & R5) even stated that one potential problem with having a modular product architecture, is that it may become harder to test, since there are many variants and dependencies. As earlier explained, one of the module drivers is the possibility to test the modules independently, which means that it should become easier to test a modular product (not harder). If the testing and/or design become harder over time, it is likely caused by sub-optimal modules, which results in a coupled design and thereby larger complexity.

Most respondents explained that the main modularisation drawback is reduced performance in some way, e.g. higher mass, higher fuel consumption etc. The cause of the reduced performance was believed to be that the complete design becomes a large compromise when everything should fit and work with everything. Since there is a high focus on increased performance today, especially in terms of mass and fuel consumption, one respondent (R3) expressed that the aim for increased performance may affect the Scania Modular System in a negative way.

According to all respondents, no specific methodology is used in the modularisation process, therefore a mixed view of how Scania modularise was identified. Respondent (R1) explained that hardware modularisation is mostly done by experience and how the gearbox has been modularised in the past. However, the goal for most respondents was to create as many product variants as possible with as few parts as possible. The strategies when modularising also involves which components that need different specification, as well as manufacturing aspects, separate testing and cost etc. However, respondent (R1) believed that it would be beneficial to use some kind of tool, where all strategies and reasons why components should become a module are specified. Most respondents also explained that they do not use operational factors when modularising.

Respondent (R1) believed that some kind of process and methodology that supports the early development of larger product changes would be beneficial, since today's work is not very structured and is therefore very dependent on the persons performing the task. The respondent said “Modules are created on the whiteboard by senior engineers, before any design in CAD”, however, the design process and design structure focus on later stages.

According to respondent (R2), the goal with software modularisation is to limit the number of software interfaces. This means that there is a low focus on the mechanical modules and the physical location of the embedded software when modularising the software. The respondent did however clarify that it will probably be harder to identify modules when the product contains more and more software in the future, due to many and complex relations between the components.

Respondent (R3) believed that it is not possible to modularise the hardware separately from everything else, therefore a holistic view is needed in order to create a good modularised product. Today, there are some general guidelines which explain where different functions should be located in the different software layers.

The result from the modularisation methodology part of the interviews clearly shows that there is no general consensus among the engineers on the preferred modularisation methodology at Scania, and that designers rely on how the modularisation has been done in the past. It also shows the difference between software and hardware designers, where software designers appear to think more in terms of system, compared to most mechanical engineers who think more about components and parts.

Since there is a large risk that complexity increases dramatically if the modules are unfavourably chosen, a methodology which supports the development of the product architecture is clearly needed at Scania. In addition to that, the methodology also needs to include how the customer needs are connected to the technical solutions, i.e. some form of a knowledge integration matrix. Finally, the strategic reasons why some components (or parts) should become a module needs to be specified.

From the interviews, it was discovered that the cable harness of the gearbox program may limit the number of product variants that can be created based on the physical modules. This shows how important it is to modularise hardware, electronics and software as a complete system. It was also discovered that hardware, electronics and software are not fully developed in parallel. This indicates that there is a functional dependency, which increases the lead time and possibly also the design time. With an ideal modular product architecture, each module could be developed and tested independently, which is not fully the case at Scania today. By applying the product architecture principle, this may be changed.

Product Description

As stated in the Frame of Reference chapter, the interfaces in a modular product architecture need to be clearly specified and documented, mainly since the interfaces are the communication point between the design teams or designers. It is therefore highly important that the interfaces are carefully chosen and specified in order to reduce complexity, and thereby reduce development time and cost. This is clearly a target area for any successful design collaboration with other companies. A large difference between how insourced and outsourced components are represented most efficiently is the usage of part numbers. A company that, as an example, has outsourced design and manufacturing of the gearboxes to external suppliers need an easy way to communicate what they have outsourced. Therefore, a unique part number is most likely created for each product variant. At Scania, which currently has mainly insourced subsystems, part numbers are not assigned to each product variant. Instead, variant code strings are used, which make it possible to describe a very large number of variants in a fairly simple way. In addition to that, the gearbox is described as a part of a system, meaning that other parameters outside of the gearbox e.g. axel distance, tyre size etc. may affect the gearbox configuration in order to gain high performance.

It should be made clear that only respondent (R1) was familiar with the design structure (KS), while respondents R4, R5 and R6 had very limited experience, and respondents R2 and R3 had never used it since they worked with embedded software. Respondents R1, R4 and R5 believed that the interfaces are poorly described in the design structure (KS), and at Scania in general. The same respondents also found it very hard to use and find information in the design structure, mainly since there is not a standardised way of working and describing the product in the structure. Respondent R4 explained that it is relatively easy to find your "own" components, however, it is a lot harder to find the interfaces to the surrounding module. The design structure does not support the cable harness design and its variants in a good way, according to respondent R4.

According to respondent R1, the design structure is used in the early phase for smaller design changes, which means that a change is done in the design structure before the actual design work is performed. However, for larger changes, e.g. when developing a new gearbox, the design structure is only used at a much later stage, when much of the design work is already done. The main reason for this is that there are extremely many changes that are constantly being made, which would be too complicated to describe in the design structure. The respondent therefore thought that some kind of other tool is needed to support the early development phase.

Respondent R1 also believed that the design structure describes the Scania Modular Toolbox. However, respondents R2, R3 and R6 were not fully sure if that is true, since they did not know if the embedded software really is a part of the Modular Toolbox.

Most respondents (4 out of 6) stated that one challenge with today's product description methodology is that it is extremely difficult to get the complete picture of all valid product variants, therefore it is e.g. hard to verify that the embedded software will work for all cases. Respondent R4 also expressed the extreme difficulties when trying to figure out which of all configurations that are valid. For a designer who only is designing one component, these combination restrictions are easy to identify, however, since the cable harness affects the complete gearbox, all valid gearbox configurations need to be identified. There is no system which contains the allowed configurations, and it is also hard to get assistance with this task, according to the respondent.

4.2 Analysis of the investigated components

The gearbox architecture at Scania is a result of a large number of small design changes over many years. This approach may be beneficial in some aspects (e.g. improved quality etc.), however, sometimes larger architecture changes needs to be made due to changes in business strategies or technology change, e.g. becoming a First-Tier supplier or electrification and digitalization of a system or subsystem. If only small changes are made over a long period of time, there is a high risk that the transformed product architecture becomes sub-optimised for the new task. It is highly important to use a robust methodology which supports the highly complex task when performing larger changes of a product architecture, otherwise important aspects may not be treated in an efficient and effective way.

As seen in Figure 68, the degree of modularity is relatively low (at the investigated level of decomposition) for the gearbox architecture, meaning that the gearbox has a more integral than modular design. However, there are still some parts which are less coupled, allowing some flexibility in order to create different product variants. These variants therefore need to be defined with configuration rules (conditions), since the modules cannot be combined arbitrarily. As stated in the Frame of Reference chapter, a fully modular product architecture should have an uncoupled design and decoupled interfaces, which would eliminate the need of most conditions. Decoupled interfaces would also allow a change to be made to one technical solution, without requiring to make a change in another technical solution, in order for the overall product to work correctly.

The gearbox architecture at Scania can thus be seen as the result of the specific modularisation process and principles used at the company during many years. Since Scania almost exclusively relies on in-sourcing, outsourcing is generally a weak module driver, compared to many other performance related module drivers. Hence, the architecture at Scania does not have to be strictly developed according to some predefined architecture rules, but rather to the solution which works best at the moment for the customer and Scania. The modules are therefore highly linked to how the product is manufactured and assembled, i.e. to manufacturing complexity, but not necessarily according to product complexity, see Figure 69.

In order to highlight some of the important aspects to consider if a modular product architecture should be successfully realised, the propeller shaft was also analysed in addition to the gearbox. Even if the architecture of the propeller shaft has a very low complexity compared to the gearbox, it shows an important feature in the Scania modular system i.e. calculating parameter values with by EXIT calculations. At some point, all different modules needs to be integrated into one system, which sometimes makes it necessary to have a module or component with a very high flexibility that is able to “take up the slack” in the system. Hence, the length of the propeller shaft is customised for each product variant depending on how other components have been configured, e.g. it depends on the axle distance.

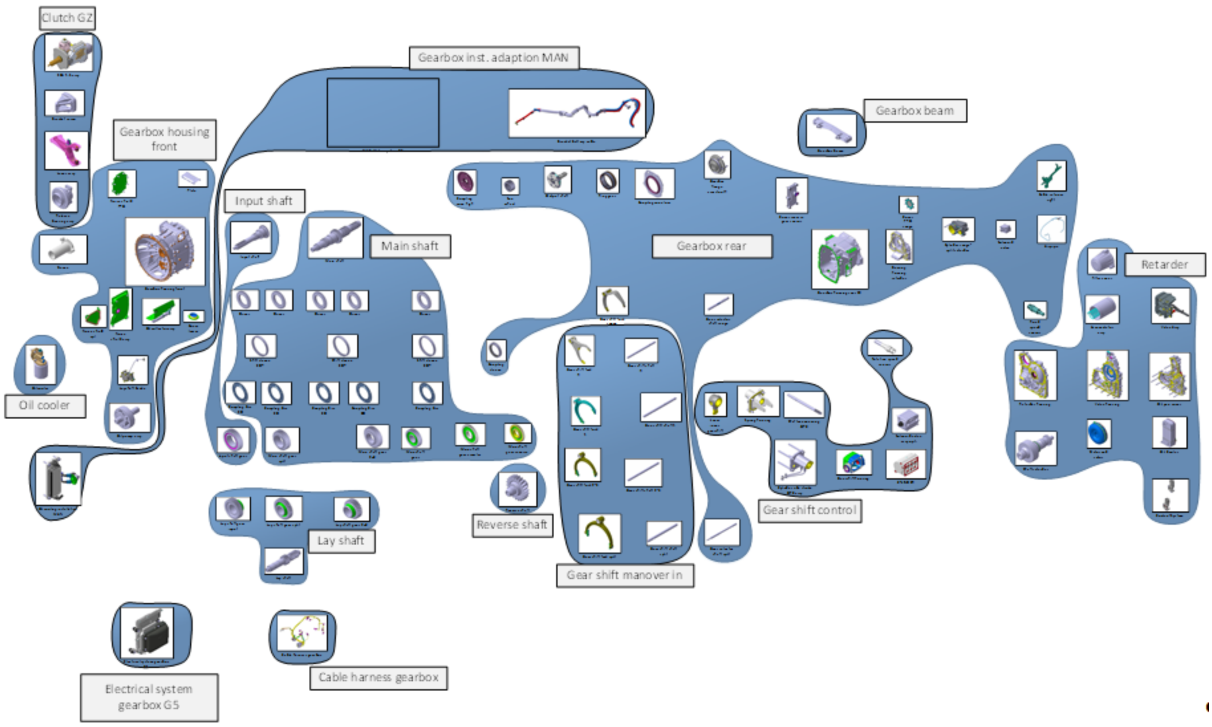


Figure 69. The modular gearbox architecture represented by a component cluster diagram (CCD).

Technical solutions that also must have very high flexibility are most cable harnesses. A cable harness therefore has many similarities to the propeller shafts in terms of how it is parametrised and manufactured for a specific vehicle. Furthermore, the embedded software is parametrised in a similar way as the propeller shaft and the cable harness, though the physical allocation to different ECU:s allows some parts of the software to be more flexible compared to a physical technical solution. As earlier defined, the embedded software is not parametrised with EXIT calculations, since this is done with the SOPS-file.

5 DISCUSSION AND CONCLUSIONS

A discussion of the results and the conclusions from the presented study are presented in this chapter.

5.1 Discussion

All types of products are not suitable to modularise, e.g. if the customers are not willing to get any reduced performance for the increase in flexibility, the product should probably not be modularised, or at least not have a high degree of modularity. For example, there have been many manufacturers (e.g. the Google Ara Project & Fairphone) trying to create a highly modular smartphone, where the customer should be able to change modules (e.g. camera module, battery module etc.). None of these manufacturers has yet been successful, simply because most customers want as high performance as possible (mainly in terms of weight, volume and water resistance etc.) and are not willing to get any reduction of performance due to the increased flexibility. This example highlights the importance of balancing the degree of modularity for a given product. In the heavy vehicle industry, the most likely outcome will probably be that a combination of different technical solutions are needed in the near future, which makes it necessary to modularise the product even more at a higher level in order to be profitable. At a lower level, the integration of more sensors, electronics and embedded software will most likely result in an even more integral design, i.e. the modules will have an integral architecture.

The investigated product architecture at Scania is clearly a hybrid between a modular and integral, meaning that it is modular at a higher level and integral at a lower level of decomposition. This hybrid architecture therefore needs to be defined with configuration rules (conditions), since the modules cannot be combined in any desired combination. Today, the modules at Scania are highly linked to how the product is manufactured and assembled. This means that product architecting and complexity management is generally a weak area, which may be a future problem when the complexity increases dramatically, due to increased amount of software and electronics. Scania will therefore need to work a lot more with product architecting and systems engineering when outsourcing becomes more and more common, i.e. going from “closed source” to “open source” in order to integrate modules from different suppliers, as well as selling modules. In the future, it will be too demanding for a single company to develop everything in-house, making these changes essential.

Due to the product description methodology and extensive development and manufacturing in-house at Scania, it has been possible to develop and manufacture a highly complex architecture, containing a high number of configuration rules. Scania therefore have limited experience if a new product architecture should be developed with robust interfaces and with low complexity, using a structured product architecting approach. If a structured methodology would be used to assist the highly complex task when developing a modular product architecture, it is highly important to not only focus on the complexity reduction, but also the strategic aspects (i.e. the module drivers) e.g. need for different performance etc. It is also highly important that the external variety never gets lowered in order to reduce complexity.

A general problem within the automotive industry is how an OEM with a high number of product variants should be able to order subsystems from a first-tier supplier, who is selling a modularised product with a high number of variants. Today, this is usually solved by ordering a sequence of part numbers for a specific product variant, making it possible to describe fairly many variants in an efficient way. However, this approach is not very flexible and cannot be used in a similar way as the variant codes at Scania.

At Scania, most of the truck modules are manufactured in-house. This means that outsourcing is generally a weak module driver, compared to many other performance related module drivers. Hence, the architecture does not have to be strictly developed according to some predefined architecture rules, but rather to the solution which works best at the moment for the customer and the company. Consequently, Scania has chosen a decentralised control strategy, i.e. much of the intelligence is embedded in the modules.

Several respondents at Scania stated that one problem with today's product description methodology is that it is difficult to get the complete picture of all valid product variants, mainly since it is only possible to verify one product variant at a time. Commercial tools are today available to support the configuration of complex modular products. For example, the Danish company *Configit A/S* offer advanced patented methodologies and tools to support the complete Configuration Lifecycle Management. These tools are similar to the Scania product description methodology in some ways, though there is a key difference. One great tool which *Configit* offer is the "Virtual Table (VT)" which makes it possible to analyse all valid configurations for a complete product portfolio in a very fast and efficient way, according to the company. This approach is a development of *Binary Decision Diagrams* which are used to solve complex constrained configuration problems. Hence, this type of tool may be highly valuable for Scania, both during the order to delivery process, but also for the engineers to find the configurations which they need to consider, e.g. when designing a cable harness to a gearbox. Many leading companies therefore use these tools, e.g. JLR (Jaguar Land Rover) and ABB etc.

The Scania methodology is based on verification of one single product variant at a time (i.e. one variant code string), which may be a fairly good solution in some aspects, though it may not be fully optimised. Since the VCR and TCR are highly integrated, and also contain *EXIT calculations*, it will most likely be highly complex to use some commercial tool without substantial amount of re-designing and re-thinking.

Furthermore, a future challenge is how the software and hardware should be described with one product description methodology. Today, the hardware and software are developed and described at different departments and with different methods, making it highly complex to investigate the complete system architecture. The future autonomous transport vehicles may be characterized as Cyber-Physical Systems (CPS) that are components of Cyber-Physical Systems of Systems (CPSoS), i.e. transport systems. A key aspect of CPS is the potential to integrate information technologies, operational technologies in terms of embedded systems and control systems, and physical electrical and mechanical systems. A drawback is that CPS significantly increase technical complexity and thus introduce new challenges to system architecting. The added complexity is preferably targeted in the product architecting development stage.

The biggest concern regarding product description was identified to be how the interfaces are described at Scania. As stated in the Frame of Reference chapter, the interfaces in a modular product architecture need to be clearly specified and documented, mainly since the interfaces are the communication points between the design teams. It is therefore highly important that the interfaces are carefully chosen and specified in order to reduce complexity. In addition to that, it needs to be a lot easier for a designer to know which of all product variants he/she should consider. By having separate structures for hardware and software, it is easy to incidentally create a coupled and thus highly complex design.

If the gearbox, which was explicitly studied, should be re-architected in the future, it is important to highlight that the choice will usually not be to design a completely modular or completely integral product architecture. The choice will rather be which system or subsystem that should be treated in a modular way, and which should be treated in an integral way. The architecture of a product is also usually dependent on the level of decomposition. At a high level, the product may have a very modular architecture, while it may have a very integral architecture at a low level. In any case, the modularisation usually needs to stop before the lowest level, otherwise the performance will usually become too low and there will be no design freedom for the designers. The complexity will also be extremely high if the product is modularised to a very low level, which will lower the product performance as well as the internal efficiency within the company. The added complexity (due to increased amount of electronics and software) is preferably targeted in a product architecting development stage.

5.2 Conclusions

The three research questions (RQ) and their condensed answers are:

RQ1: What is the present state at Scania, regarding product architecture and management of product data?

- The interviews clearly indicate that there is no general consensus among the design engineers on the nomenclature and terminology, e.g. function, product architecture, product structure, modular architecture, standardisation, etc. In the field of mechanical engineering, the word *product property* is sometimes thought to be the same thing as a *function*, which is an unfortunate misconception. The terminology clearly needs to be defined more rigorously at Scania, preferably with a definition which is consistent and aligned with accepted terminology from design theory and systems engineering in order to reduce the risk of confusion and design mistakes internally at Scania and especially when collaborating with external partners. A consistent nomenclature also makes it easier for everyone to aim for the same target.
- The investigated gearbox is described as a part of a Scania system, meaning that other parameters outside of the gearbox e.g. axel distance, tyre size etc. may affect the gearbox configuration. This approach makes the product description at Scania partly integral.
- Configuration rules (conditions) are highly important in order to successfully realise a modular product architecture. The conditions at Scania are located at all levels in the product structures, which allows the product to be configured in a highly complex way. A drawback with this approach is that the solution space (i.e. all valid configurations) becomes extremely challenging to identify.
- EXIT-calculations are Scania unique and allow some components to be efficiently tailor-made for each product variant. A good understanding of manufacturing aspects must also be considered when identifying the parametrised components.

RQ2: What are the unique properties in the modular product architecture at Scania and how are they used, developed and maintained?

- Based on the results from the interviews it is clear that Scania has a clear business strategy for the product architecting stage, and a fairly modular product architecture at a high level. The modularisation strategy at Scania is based on a sort of informal consensus that parts add cost, and variants add opportunities. As a consequence, Scania aims at creating as many product variants as possible to provide variety to the customers, with a limited number of parts to satisfy the requirement of internal commonality.

- Scania strives to maximise the number of product variants (external variety), while keeping the number of technical solutions low (internal commonality). Furthermore, the product variants are not designed to satisfy some predefined and specific customer requirements and during configuration they are chosen as late as possible (late variant definition) when the actual customer demand is known.
- The modules at Scania are highly linked to how the product is manufactured and assembled, i.e. to manufacturing complexity, but not necessarily according to product complexity.
- At Scania, most of the truck modules are manufactured in-house. This means that outsourcing is generally a weak module driver at Scania, compared to many other performance related module drivers. Hence, the architecture does not have to be strictly developed according to some predefined architecture rules, but rather to the solution which works best at the moment for the customer and the company. Scania has chosen a mainly decentralised control strategy architecture, i.e. much of the intelligence is embedded in modules, and much of the higher order control functions are also distributed, and thus taking advantage of computing power wherever it is available.

RQ3: How can we represent the product architecture in general and a modular architecture in particular in a way that facilitates cross-functional communication and collaboration on architecture-related tasks?

Discussion with engineers and domain experts has shown that communication and design analyses can be enabled with three different and complementary architectural representations:

- The component structure and the functional relations can represented in a very condensed form as a *Product Architecture DSM*, which is a component DSM, but with the dependencies represented as functional flows (energy, matter and/or information flow) and/or spatial relations. This representation can be used for product clustering analyses.
- A graphical representation, referred to as *Component Architecture Diagram* (CAD), has been created and used in the project. The new representation was highly appreciated by the domain experts, and the CAD showed to make architectural discussions in general and modularity discussion in particular with and between domain experts efficient.
- The Component Cluster Diagram (CCD), which is a simplification of the CAD since it represents the modular clusters but with no interactions represented, showed to be an efficient tool for presenting the modular view of the architecture.

6 RECOMMENDATIONS AND FUTURE WORK

Finally, after conducting the project, the author has identified the following work to be interesting and important to study further.

- Define a consistent nomenclature at Scania and align it with the terminology at Volkswagen Truck & Bus GmbH, in order to reduce the risk of confusion and design mistakes during the collaboration between Scania, MAN and other potential future brands.
- The designers would probably need more education within the area of product architecting in order to design according to predefined architecture goals.
- Investigate how the embedded software, electronics and mechanical hardware may be described in one unified product architecture representation.
- Use a systematic product architecting methodology when representing, analysing and developing the modular system and its constituents, i.e. the modules. Furthermore, the methodology also needs to include how the customer requirements are connected to the technical solutions, i.e. some form of a knowledge integration map/matrix. Finally, the strategic reasons why some components (or parts) should become a module must be made explicit.
- Several respondents stated that one problem with today’s product description methodology is that it is extremely difficult to get the complete picture of all valid product variants, therefore it is e.g. hard to verify that the embedded software will work for all cases. For a designer who only is designing one component, these combination restrictions are easy to identify, however, since e.g. the cable harness affects the complete gearbox, all valid gearbox configurations need to be identified with the support from some new system analysis tool.
- At Scania, there is a strong need of support from the product description in the early product development phase. A study of how the product architecture should be represented and how the different domain-specific representation should be integrated/mapped is proposed.
- A new tool which supports the early development of larger product changes is also clearly needed. Finally, an investigation targeting if and how the embedded software should be a part of the design structure must be performed. By having separate structures for hardware and software, it is easy to incidentally create a coupled and thus highly complex design.
- To enable the designers to see the complete picture and follow the principle “same need, identical solution”, the Scania modular toolbox should visualised in a clear and easy form.

7 REFERENCES

Backman, J., 2016. *Rapporter och uppsatser.* 3d ed. s.l.:Studentlitteratur.

Blackenfelt, M., 2001. *Managing complexity by product modularisation,* Stockholm: Royal Institute of Technology.

Boeing, 2015. *boeing.com.* [Online]
Available at: http://www.boeing.com/boeing/commercial/777family/background.page?

Brecher, C., 2012. *Integrative Production Technology for High-Wage Countries.* Berlin: Springer.

Dieter, G. E. & C, S. L., 2012. *Engineering Design.* 5th ed. New York: McGraw-Hill.

Eppinger, S. D. & Browning, T. R., 2012. *Design Structure Matrix Methods and Applications.* s.l.:The MIT Press.

Eppinger, S. D., Whitney, D. E., Smith, R. P. & Gebala, a. D. A., 1994. A Model-Based Method for Organizing Tasks in Product Development. *Springer.*

Erixon, G. & Ericsson, A., 1999. *Controlling Design Variants.* s.l.:Society of Manufacturing Engineers.

Foussier, P., 2006. *From Product Description to Cost: A Practical Approach.* Volume 1: The Parametric Approach ed. London: Springer.

Göpfert, J., 1998. *Modulare Produktentwicklung: Zur gemeinsamen Gestaltung von.* Wiesbaden: Deutscher Universitäts-Verlag.

Hölttä-Otto, K., 2005. *MODULAR PRODUCT PLATFORM DESIGN,* Helsinki: Helsinki University of Technology.

Kossiakoff, A., Sweet, W. N. & Seymour, S., 2011. *Systems Engineering Principles and Practice.* 2nd ed. s.l.:Wiley.

Kratochvíl, M. & Carson, C., 2005. *Growing Modular.* Berlin: Springer.

Kvale, S., 1997. *Den kvalitativa forskningsintervjun.* Lund: Studentlitteratur.

Robotham, A., 2002. *The use of function/means trees for modelling technical, semantic and business functions,* s.l.: Journal of Engineering Design.

Saaksvuori, A. & Immonen, A., 2008. *Product Lifecycle Management.* Third Edition ed. Berlin: Springer.

Scania, 2009. *Modularisation - A Way of Thinking.* Södertälje: Scania CV AB.

Scania, 2017. *scania.com.* [Online]
Available at: https://www.scania.com/se/sv/home/experience-scania/features/the-man-behind-the-modular-system.html
[Accessed 28 03 2017].

Silverstein, D., Samuel, P. & Decarlo, N., 2009. *The Innovator's Toolkit: 50+ Techniques for Predictable.* s.l.:Wiley .

Simpson, T. W., Jiao, J. R. & Siddique, Z., 2014. *Advances in Product Family and Product platform Design, Methods & Applications.* New York : Springer.

Simpson, T. W., Siddique, Z. & Jiao, J. R., 2006. *Product Platform and Product Family Design.* New York: Springer.

Sköld, M., 2016. *Dubbelt effektiv.* 1 ed. s.l.:Rheologica Publishing.

Ullman, D. G., 2010. *The Mechanical Design Process.* Fourth Edition ed. s.l.:Mc Graw Hill.

Ulrich, K., 1993. The role of product architecture in the manufacturing firm. *Elsevier.*

Williamsson, D. & Sellgren, U., 2016. An approach to integrated modularization. *Elsevier,* Volume 50, pp. 613 - 617.

APPENDIX: Semistructured interview guide

Questions

General (2 min)

1. Years at Scania?
2. Years at the current department?
3. Position, e.g. Mechanical, Electric or Software design engineer?

Nomenclature (10 min)

1. Do Scania have modules?
2. What is a module? Is there any difference between a module and a component series?
3. Describe a function.
4. Describe a product property.
5. What is a product architecture?
6. Have you identified any communication problem at Scania, due to insufficient definition of the nomenclature? If yes, please explain when and how that happened?

Modularisation (20 min)

1. What is the opposite to a modular product architecture?
2. Is it acceptable for one function to be realised in different components?
3. Should all products in the world be modularised? Why/why not?
4. Is the Scania product fully modular? Which parts are /are not modular?
5. What are the drawbacks with modularisation?
6. Is standardisation the same thing as modularisation? In which way are they different?
7. Do you have the support and methods that you need during the modularisation stage?
8. How do you modularise according to the customer needs?
9. Do you focus extra on some function/department within Scania when modularising, e.g. R&D, Manufacturing, Purchase or After-sales? In which way?
10. Is it harder to modularise when the product contains more electronics and software?
11. How do you use the operational factors when you modularise?
12. Is it possible to see how a product property or a customer need is linked to a technical solution? If no, would it be beneficial?

Product Description (20 min)

1. Are the interfaces clearly documented and easy to access?
2. Is it easy to get a complete picture of the Scania Modular Toolbox?
3. Is the design structure (KS) the same as the Scania Modular Toolbox?
4. Does the design structure support the Scania modularisation principles?
5. How do you know in which configurations your component/module is used and valid?
6. Does the product description support your work during the design? In which way?
7. If you would describe your component, would you describe what is does or what it is?